English

FUJITSU

openFT V12.0 for BS2000/OSD

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 . . . 9

1.1 Brief description of the product openFT for BS2000/OSD . . . 10
1.2 Target group and objectives of this manual . . . 10
1.3 Concept of the openFT for BS2000/OSD manuals . . . 11
1.4 Changes since the last version of the manual . . . 12
1.5 Notational conventions . . . 14
1.6 README file . . . 15
1.7 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 The OSI reference model . . . 24
2.2.2 Position of the openFT product family in the OSI Reference Model . . . 26
2.2.3 openFT partners . . . 27
2.2.4 FTAM partners . . . 27
2.2.5 FTP partners . . . 28

2.3 Transferring files . . . 31
2.3.1 Specifying the transfer start time . . . 32
2.3.2 Controlling the duration of a request . . . 32
2.3.3 Request queue . . . 33
2.3.4 Automatic restart . . . 34

2.4 File management . . . 35
2.5 Remote command execution . . . 36

2.6 Automation . . . 37
2.6.1 File transfer with preprocessing, postprocessing and follow-up processing . . . 37
2.6.1.1 Preprocessing . . . 38
2.6.1.2 Postprocessing . . . 38
2.6.1.3 Follow-up processing . . . 38
2.6.2 Program interfaces . . . 40
2.6.3 Job variables in BS2000 . . . 40

2.7 Further processing of openFT data . . . 41

2.8 Secure operation . . . 42
2.8.1 The FTAC function . . . 42
2.8.1.1 Features of the FTAC function . . . 42
2.8.1.2 Admission set . . . 44
2.8.1.3 FT profile (admission profile) . . . 44
2.8.1.4 Effects of an admission profile . . . 47
2.8.1.5 FTAC administrator . . . 48
2.8.2 Encryption for file transfer requests . . . 48
2.8.3 Logging openFT operations - the logging function . . . 50
2.8.4 Authentication . . . 52

2.9 Using openFT in a HIPLEX-composite . . . 54

3 File transfer and file management . . . 55

3.1 File names . . . 56
3.1.1 Unique file names for receive files . . . 56
3.1.2 BS2000/OSD file names . . . 58
3.1.3 File names in Unix systems . . . 60
3.1.4 Windows file names . . . 60
3.1.5 z/OS file names . . . 61

3.2 File passwords . . . 64

3.3 File types . . . 65
3.3.1 BS2000/OSD files . . . 65
3.3.1.1 Creating files in BS2000 . . . 66
3.3.1.2 Transferring library members . . . 67
3.3.2 z/OS files . . . 68
3.3.3 Unix and Windows files . . . 69
3.3.4 FTAM files . . . 71
3.3.5 Transfer of various file types . . . 72
3.3.6 Migrated files . . . 76

3.4 Transferring 7-bit, 8-bit and Unicode files . . . 77
3.4.1 Code tables and coded character sets (CCS) . . . 77
3.4.2 Specifying the CCS on a transfer request . . . 78
3.4.3 Data conversion . . . 79
3.4.4 XHCS support by openFT . . . 80
3.4.4.1 Binary file transfer . . . 80
3.4.4.2 File transferred as text file . . . 80

3.5 Entries for the remote system . . . 84
3.5.1 Defining the partner computer . . . 84
3.5.2 Transfer admission . . . 88

3.6 Options for file transfer . . . 91
3.6.1 Maximum record lengths . . . 91
3.6.2 Syntax rules . . . 91
3.6.3 Compressed file transfer . . . 94
3.6.4 Encrypted file transfer . . . 94
3.6.5 Transfer of protection attributes . . . 95
3.6.6 Notifying results . . . 95
3.6.7 Access mode . . . 95
3.6.8 Preprocessing and postprocessing . . . 96
3.6.9 Follow-up processing . . . 99
3.6.10 Monitoring using job variables . . . 101

3.7 File management . . . 103
3.7.1 File management in the remote system . . . 103
3.7.2 File management in the local system . . . 104

3.8 Special points for file transfer with FTAM partners . . . 106
3.8.1 Virtual filestore . . . 106
3.8.2 Mapping FTAM attributes to the real file system . . . 109
3.8.2.1 Inbound mapping of FTAM attributes . . . 109
3.8.2.2 Inbound mapping the document type . . . 111
3.8.2.3 Access protection . . . 115
3.8.2.4 Outbound mapping of the document type . . . 116
3.8.3 Addressing via Application Entity Title (AET) . . . 126

4 Command interface . . . 129

4.1 Functional command overview . . . 130
4.1.1 FT command overview . . . 130
4.1.2 FTAC commands overview . . . 132

4.2 Entering FT commands . . . 133

4.3 Command syntax representation . . . 135
4.4 Command return codes . . . 153
4.5 OPS variables . . . 154
4.6 Output in CSV format . . . 155
4.7 CANCEL-FILE-TRANSFER
Cancel file transfer requests . . . 156
4.8 CREATE-FT-PROFILE
Create admission profile . . . 163
4.9 CREATE-REMOTE-DIR
Create remote directory . . . 185
4.10 DELETE-FT-PROFILE
Delete admission profile . . . 188
4.11 DELETE-REMOTE-DIR
Delete remote directory . . . 191
4.12 DELETE-REMOTE-FILE
Delete remote files . . . 194
4.13 EXECUTE-REMOTE-CMD
Execute remote command . . . 197
4.14 MODIFY-FILE-FT-ATTRIBUTES
Modify local FT file attributes . . . 202
4.15 MODIFY-FILE-TRANSFER
Modify request queue . . . 207
4.16 MODIFY-FT-ADMISSION-SET
Modify admission set . . . 213
4.17 MODIFY-FT-PROFILE
Modify admission profile . . . 219
4.18 MODIFY-REMOTE-DIR-ATTR
Modify remote directory attributes . . . 244
4.19 MODIFY-REMOTE-FILE-ATTRIBUTES
Modify remote file attributes . . . 247
4.20 SET-FT-INSTANCE
Set an openFT instance . . . 255
4.21 SHOW-FILE-FT-ATTRIBUTES
Display local FT file attributes . . . 256

4.22 SHOW-FILE-TRANSFER
Query status of file transfer request 260
4.22.1 Description of the short output 271
4.22.2 Description of the long output 272
4.22.3 Description of the summary output 276

4.23 SHOW-FT-ADMISSION-SET
Display admission sets 277

4.24 SHOW-FT-INSTANCE
Display an openFT instance 282

4.25 SHOW-FT-LOGGING-RECORDS
Display log records and offline log files 284
4.25.1 Description of the short output 294
4.25.2 Description of the long output 296

4.26 SHOW-FT-MONITOR-VALUES
Show monitoring data 303
4.26.1 Description of the monitoring values 309
4.26.2 Examples 314

4.27 SHOW-FT-OPTIONS
Display operating parameters 317
4.27.1 Description of the output 323

4.28 SHOW-FT-PARTNERS
Display partner systems 328

4.29 SHOW-FT-PROFILE
Display admission profile 337

4.30 SHOW-FT-RANGE
Display partner systems 344

4.31 SHOW-REMOTE-FILE-ATTRIBUTES
Display remote file attributes 347
4.31.1 Description of the output 352
4.31.2 Description of output fields 352

4.32 TRANSFER-FILE
Transfer file asynchronously 361
4.32.1 Introduction to the TRANSFER-FILE command 361
4.32.1.1 The shortest form of the command 362
4.32.1.2 How to find out if the file transfer request has been executed 362
4.32.2 Full form of the TRANSFER-FILE command 363
Specifications for the local system (LOCAL-PARAMETER) 368
Follow-up processing in the local system 374
Specifications for the remote system (REMOTE-PARAMETER) 378

Follow-up processing in the remote system . . . 388
FTAM-specific operands . . . 392
Optional entries . . . 395
4.32.3 Examples of the TRANSFER-FILE command . . . 403

4.33 TRANSFER-FILE-SYNCHRONOUS
Transfer file synchronously . . . 414

5 What to do if... . . . 421

5.1 Frequently asked questions . . . 428

6 Appendix . . . 433

6.1 Command return codes for file transfer and file management . . . 433

6.2 Structure of CSV outputs . . . 440
6.2.1 Output format . . . 440
6.2.2 SHOW-FILE-TRANSFER . . . 441
6.2.3 SHOW-FT-ADMISSION-SET . . . 446
6.2.4 SHOW-FT-LOGGING-RECORDS . . . 448
6.2.5 SHOW-FT-MONITOR-VALUES . . . 451
6.2.6 SHOW-FT-OPTIONS . . . 455
6.2.7 SHOW-FT-PARTNERS . . . 460
6.2.8 SHOW-FT-PROFILE . . . 462
6.2.9 SHOW-FT-RANGE . . . 466
6.2.10 SHOW-REMOTE-FILE-ATTRIBUTES
SHOW-FILE-FT-ATTRIBUTES . . . 467

Glossary . . . 469

Abbreviations . . . 495

Related publications . . . 501

Index . . . 505

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 BS2000/OSD

openFT for BS2000/OSD is the file transfer product for computers using the operating system BS2000/OSD.

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 allows the use of TCP/IP, ISO TP0/2, ISO TP4, SNA and NEA as transport protocols.

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

- openFT also supports the FTAM file transfer protocol (File Transfer Access and Management) standardized by ISO (International Organization for Standardization). This makes it possible to interconnect to systems of other manufacturers whose FT products also support the same standard.
- 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 and objectives of this manual

This manual is intended for BS2000/OSD 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 the functions provided byopenFT-AC.

To understand this manual, it is necessary to have a knowledge of the BS2000 operating system.

1.3 Concept of the openFT for BS2000/OSD manuals

The complete description of openFT and its optional components openFT-FTAM for BS2000, openFT-FTP for BS2000 and openFT-AC for BS2000 is contained in three manuals. In addition to this User Guide, there is also a System Administrator Guide and a Programmer Reference Guide. The description is divided between the three manuals as follows:

- openFT for BS2000 - Installation and Administration

  The System Administrator Guide is intended for FT and FTAC administrators. It describes:
  - the installation of openFT and its optional components
  - operation, control and monitoring of the FT system and the FTAC environment
  - the administration commands for FT and FTAC administrators
  - account records

- openFT for BS2000 - 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 the file transfer to computers with different operating systems
  - information on the implementation of FTAM
  - description of the user commands
  - messages from openFT and openFT-AC

- openFT for BS2000 - Programming Interfaces

  The Programmer Reference Guide describes the openFT and openFT-AC program interfaces.

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

The following changes have been introduced in the openFT V12.0 for BS2000/OSD User Guide since the earlier version openFT V11.0 for BS2000:

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 SHOW-FT-LOGGING-RECORDS 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 SHOW-FT-LOGGING-RECORDS, 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 SHOW-FT-LOGGING-RECORDS, 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. SHOW-FT-LOGGING-RECORDS 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 SHOW-FT-PARTNERS, 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 SHOW-FT-PARTNERS, 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 SHOW-FILE-TRANSFER and SHOW-FT-LOGGING-RECORDS 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.

Default account number for users

If ACCOUNT=*NONE is specified then a user's default account number is used. ACCOUNT=*NONE can be specified for the following commands and operands:

- CREATE-FT-PROFILE and MODIFY-FT-PROFILE, operands USER-ADMISSION and PROCESSING-ADMISSION.
- TRANSFER-FILE and TRANSFER-FILE-SYNCHRONOUS, operands TRANSFER-ADMISSION and PROCESSING-ADMISSION

Other changes

- Take over modification time of send file
- By specifying the operand LAST-CHANGE-DATE in the commands TRANSFER-FILE and TRANSFER-FILE-SYNCHRONOUS, it is possible to control whether the modification time of the send file is taken over into the receive file.
- The description of the OPS variable in SHOW-FT-PARTNERS has been extended to include the output parameters for ADM partners.
- 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 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 135.

1.6 README file

The functional changes to the current product version and revisions to this manual are described in the product-specific Readme file.

Readme files online

Readme files are available to you online in addition to the product manuals under the various products at *http://manuals.ts.fujitsu.com*.

Readme files under BS2000/OSD

On your BS2000 system you will find Readme files for the installed products under the file name:

```
SYSRME.OPENFT.120.E
SYSRME.OPENFT-FTAM.120.E
SYSRME.OPENFT-FTP.120.E
SYSRME.OPENFT-AC.120.E
```

Please refer to your system administrator for the user ID under which the required Readme file can be found. You can also obtain the path name of the Readme file directly by entering the following command:

```
/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=<product>,LOGICAL-ID=SYSRME.E
```

You can view the Readme file on screen with `/SHOW-FILE` or by opening it in an editor, or print it on a standard printer using the following command:

```
/PRINT-DOCUMENT <filename>, LINE-SPACING=*BY-EBCDIC-CONTROL
```

Additional product information

Current information, version and hardware dependencies, and instructions for installing and using a product version are contained in the associated Release Notice. These Release Notices are available at *http://manuals.ts.fujitsu.com*.

1.7 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 aheared 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 rouines 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, an FTAM product 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 77.

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.

2.2.1 The OSI reference model

In order to exchange data, systems must be able to intercommunicate. Communication is possible only if the computers involved use the same file formats for data exchange and observe an agreed behavior during transfer. The sum of the conventions and file formats for communication is referred to as a protocol. Protocols are defined by the manufacturer (for example openFT protocols) on the one hand, and on the other by committees which define manufacturer-independent protocols. ISO (International Organization for Standardization) provides the OSI Reference Model (**O**pen **S**ystems **I**nterconnection), the best-known model for communications architecture and the most comprehensive collection of protocols.

The OSI Reference Model structures the communications functions of computer systems and provides a foundation for standardization of protocols and services. It specifies which functions the components involved in communication must provide.

The OSI Reference Model consists of seven hierarchically structured layers. Each layer is assigned specific communication functions.

Layers	Designation	Functions	
Layer 7	Application Layer	Coordinates and controls the performance of communication tasks for an application	**A**PPLICATION
Layer 6	Presentation Layer	Regulates the form of information presentation and thus permits user/device-independent communication	
Layer 5	Session Layer	Regulates the sequence of communication	
Layer 4	Transport Layer	Regulates the reliable exchange of data between two communications partners	**T**RANSPORT
Layer 3	Network Layer	Regulates the exchange of data between two terminal systems (computers)	
Layer 2	Data Link Layer	Secures the transmission on individual subroutes of the entire transmission route (procedures)	
Layer 1	Physical Layer	Provides the physical connection (via the medium used for transmission)	

OSI Reference Model

The individual layers use the service of the layer immediately below and provide a precisely defined service to the layer above. Only the physical layer must provide its service together with the physical medium. The active elements within a layer, which provide the functions, are referred to a instances.

Each layer is specified by the service it provides, and the services it uses from the layer below it. During communication, the various computers interoperate on the same layer, using common protocols.

The functionality of each layer in the OSI Reference Model can be provided by various protocols as a rule. Decisive for the communication is that the direct partner instances use the same protocol for a particular task. For this purpose, profiles are defined.

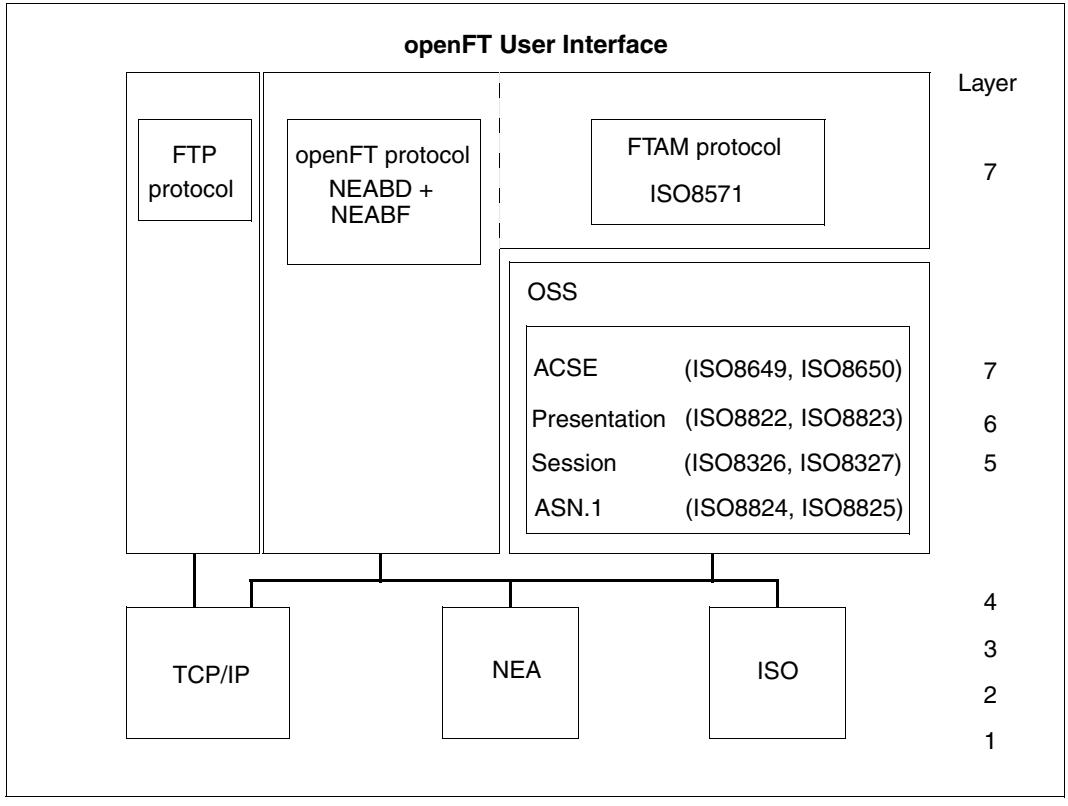
A profile is understood as precise specification of which protocols or which protocol variants are to be used on which layer to perform a particular task. Profiles are stipulated by national or international organizations or communities.

2.2.2 Position of the openFT product family in the OSI Reference Model

The openFT products belong to the application layers (Layers 5 - 7) of the OSI Reference Model. They support the standardized openFT protocol and the FTAM protocol ISO8571 standardized by ISO and the File Transfer Protocol (FTP) defined by RFC959.

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

The following diagram shows the possible combinations of application and transport protocols for file transfer:



Protocols supported by openFT in the environment of the OSI Reference Model

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

2.2.3 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
– NEA transport protocols
– ISO 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.

2.2.4 FTAM partners

The FTAM extension available in openFT also enables openFT to perform file transfer and file management with partner systems which support ISO protocols in layers 5 - 7 of the OSI Reference Model. In the rest of this manual, these systems are referred to as FTAM partners, since they use the protocols for file transfer defined in the international standard ISO 8571 (FTAM, File Transfer, Access and Management).

BS2000/OSD also require the OSS software package to implement layers 5 - 7.

Implementation of FTAM Standards in openFT

A subset of the complete functional scope of the base standards has been selected in accordance with international and European profiles ISO/EN ISP 10607-3 and ISO/EN 10607-6. This functional standardization has, in turn, been harmonized with other functional standards (and implementation agreements), e.g. the corresponding implementation agreements of IGOSS in North America and corresponding profiles in Asia and Australia.

ENV 41204 and ENV 41205 are the old, nevertheless still applicable, designations for EN 10607-3 and EN 10607-6 and their contents are identical to the international profiles ISO/IEC ISP 10607-3 (1990) and ISO/IEC ISP 10607-6 (1990) agreed by ISO. EN 10607-3 and EN 10607-6 contain additional European character repertoires.

These profiles specify the file attributes actually used, for example, and the operations permitted with these attributes, irrespective of the operating system used. A **virtual filestore** is used to permit presentation across several operating systems; here, the contents of the real store are transferred with a representation of the file attributes in accordance with the standard. Conversion of the file attributes to FTAM Standard in the operating system and vice versa is part of the FTAM functionality. There are three groups of file attributes: kernel group, storage group and security group (see page 106).

Compliance with the FTAM standard also restricts the functional scope offered by openFT protocols. Transfer of follow-up data to FTAM partners is not possible with the protocol.

The mapping mechanism between the real filestore and the virtual filestore is described in detail on page 106.

2.2.5 FTP partners

Alongside openFT and FTAM 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 65 and section “z/OS files” on page 68. 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 37.

You should not process a file further until transfer is completed; otherwise, inconsistencies may result. DMS files are protected during the entire execution of the request.

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 32). 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 50.

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.

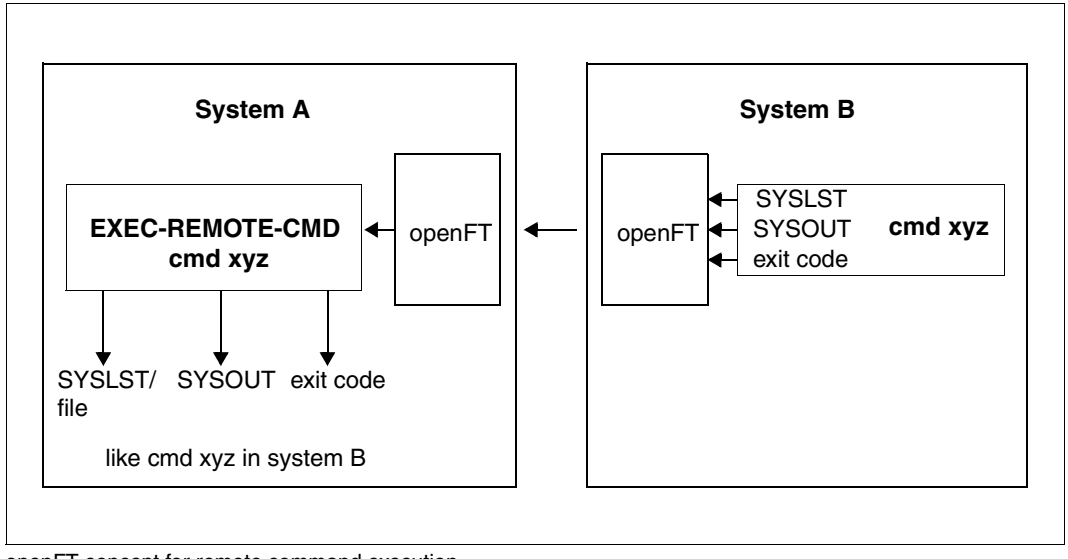
You can perform the following actions with via file management:

– 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.

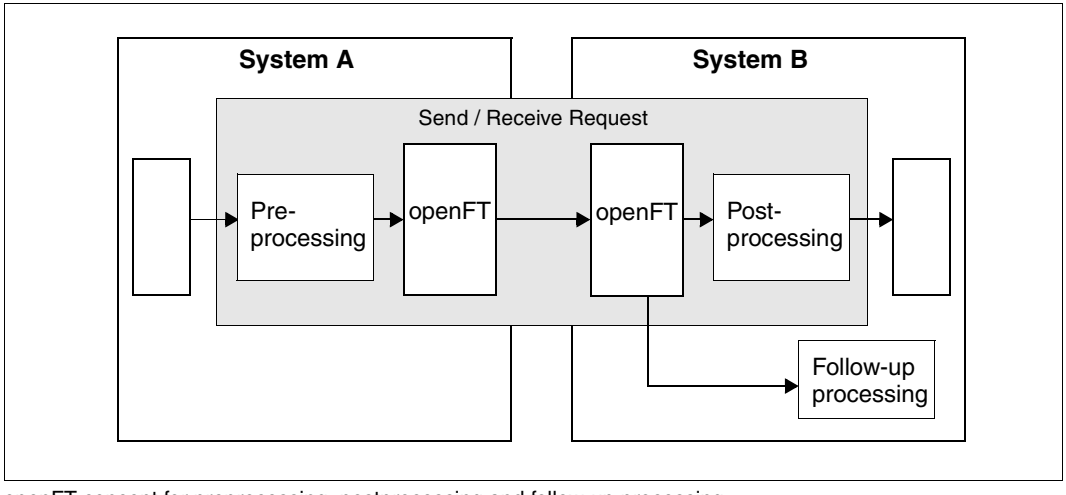
2.6.1 File transfer with preprocessing, postprocessing and follow-up processing

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.

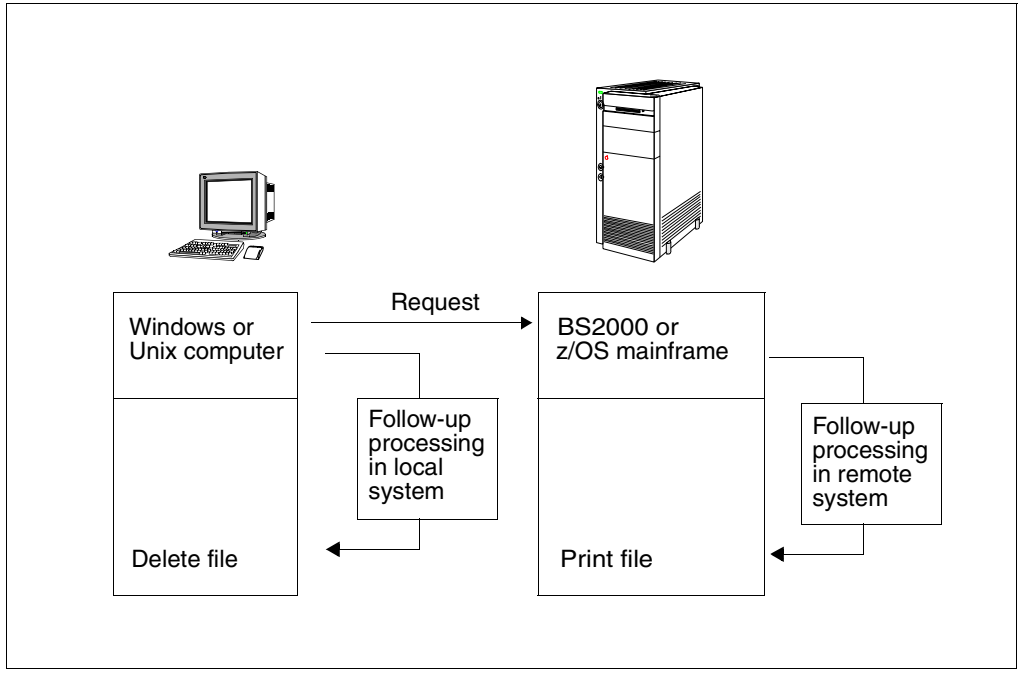
openFT requests with remote preprocessing or postprocessing can also be transferred by older versions of openFT or FT. It is important that a version of openFT that supports postprocessing is used in the remote system.

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.

2.6.3 Job variables in BS2000

openFT for BS2000/OSD offers the option of using a job variable to monitor a file transfer request. The name of the job variable, as well as any password required to access this variable are specified with the file transfer request. The job variable that monitors a request is also used to identify the request and may be used as a selection criterion to abort the request or obtain information about it. Further, it can be used for event control purposes, or to change request priorities.

2.7 Further processing of openFT data

In order to permit openFT data (SHOW-FT-LOGGING-RECORDS, SHOW-FT-OPTIONS, 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.

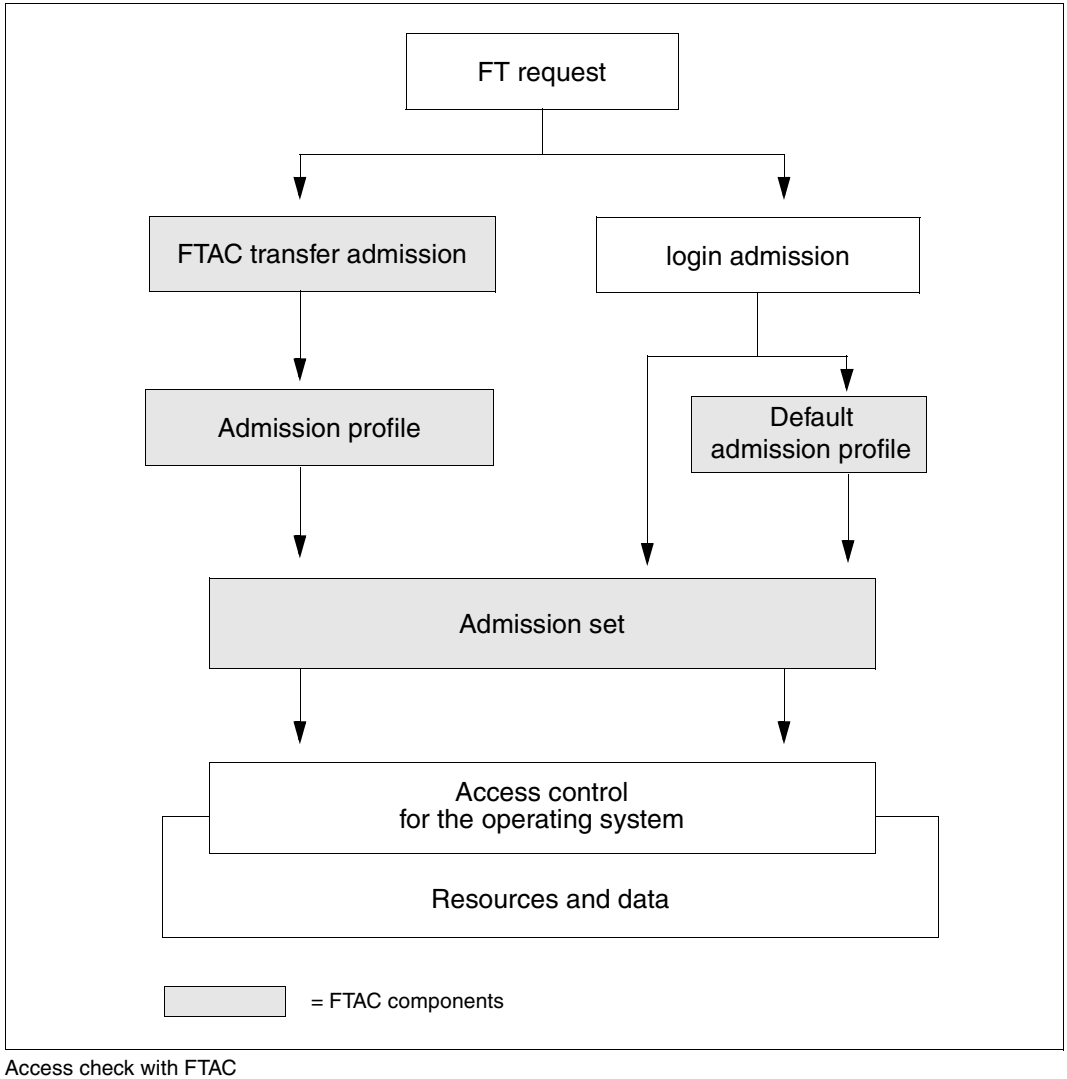
In addition, the FT administrator can assign security levels to the partner systems. When combined with the admission set settings, this makes it possible to prohibit or permit the use of the individual file transfer functions on a partner-specific basis.

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 47. 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.

  For example, if you specify PREFIX = USER in BS2000 then an FT request in which FILE-NAME = HUGO has been specified will access the file USER.HUGO.

  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.

In BS2000/OSD, the product openCRYPT is used for encryption, assuming it is installed and has been started. Otherwise, the openFT internal encryption algorithms are used (as done previously).

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 41).

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 SHOW-FT-LOGGING-RECORDS

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:

SYSPKF.R<key reference>.L<key length> in the configuration user ID of 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 IMPORT-FT-KEY to import a partner system's public key.

2.9 Using openFT in a HIPLEX-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 use the command SET-FT-INSTANCE

Furthermore, you can output information on the instances with the SHOW-FT-INSTANCE 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). If a partner using the FTAM functionality is involved in file transfer, the link is a heterogeneous one as a rule. 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 42), 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 44). 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.

%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 TRANSFER-FILE command with WRITE-MODE = *REPLACE-FILE or *NEW-FILE

– file names in CREATE-FT-PROFILE or MODIFY-FT-PROFILEto preset the receive file name in FTAC profiles

The following restrictions must be observed when using %UNIQUE:

– The metacharacter string %UNIQUE can only be used when specifying a file name as a C-string. The file name specified here may be a DMS file, a POSIX file or a library member.

– The string %UNIQUE may be preceded by a prefix, provided the syntax rules for file names (especially the maximum file length) are not violated after %UNIQUE is replaced by the defined string (see above).

- % characters are allowed in POSIX file names. Consequently, if the string %UNIQUE or %unique appears (anywhere) in a file name, and no substitution is desired, the string must be entered in the command as %%UNIQUE or %%unique. The duplicate % character will be removed on the target system.
- %UNIQUE can be used to uniquely identify a library member, but may only be specified in the member name or the version designation, not both. The member cannot be entered using the *LIB(...) syntax in this case. The following syntax options are available:
  - C'lib/type/prefix%UNIQUE(version)'
  - C'lib/type/prefix%UNIQUE()/prefix%UNIQUE(version)'
  - C'lib/type/member(prefix%UNIQUE)'
  - C'lib/type/member()/member(prefix%UNIQUE)'

Note that in log records, result logs, the output of the SHOW-FILE-TRANSFER 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.

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 local and /or 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.

If files are transferred using POSIX commands, local DMS names must be specified including the catid. Names specified without the catid are interpreted as POSIX files. The prefix "::" can be used for the default catid.

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".

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/.

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.

3.3.1.1 Creating files in BS2000

If the receive file is not yet present then it is set up by openFT for BS2000. When library members are received, any libraries that are not present are also implicitly set up. The file or library cannot be set up under the following circumstances:

– if the file belongs to a file generation group that has not yet been cataloged.

– if the file or library member is specified in fully qualified form and the user ID present in the name does not match the user ID to which the transfer admission refers (possibly via an FTAC admission profile) unless the transfer admission refers to $TSOS

Library members that are not yet present are also implicitly set up. If the member name does not specify a version then the highest possible version number is assumed. The specification WRITE=NEW in the FT request protects existing members in the receive library with TYPE, NAME and VERSION definitions against being overwritten.

A receive file that has been newly set up by openFT for BS2000 or an implicitly created receive library possesses the following file protection features:

– no file password
– write and read access permitted
– not shareable and
– no period of protection against modification

Special receive file properties:

- If a file is transferred to an NK pubset using BLKCTRL=PAMKEY then it is implicitly converted to BLKCTRL=DATA. In this case, openFT for BS2000 offers the same functionality as the PAMCONV utility (see the “Utility Routines (BS2000)” manual).

  PAMKEY cannot convert files with BLKSIZE (STD,16) to DATA.

  In the case of transfer to an NK4 pubset it may be necessary to perform conversion with PAMCONV. If no conversion is performed then only evenly blocked files can be transferred to NK4 pubsets. ISAM files must additionally possess an NK4 format (BLKCTRL=DATA(NK4)).

- If the receive file is password-protected and if the buffer size of the send file BLKSIZE ≥ 2 (PAM blocks) then the primary allocation of the send file must be sufficiently large. The required primary allocation is dependent on the BS2000 version and can be found in the current DMS manuals.

- The secondary allocation for receive files that did not previously exist or are not password-protected has, following transfer, approximately the value

  ```
  Size of send file / 4.
  ```

  The secondary allocation of password-protected receive files that have already been set up remains unchanged.

- PAM files with empty blocks (coded file-id on OPEN ≠ coded file-id of the PAM block) are transferred together with the empty blocks if the number of successive empty blocks satisfies the following conditions:
  - Receive file not set up or not password-protected:
    ```
    Number of empty blocks ≤ (M / 4) – 3,
    ```
    where M = 72 or is equal to the block number of the last PAM block written in the send file if this block number > 72.
  - Receive file is password-protected:
    ```
    Number of empty blocks ≤ secondary allocation of receive file
    before file transfer – 3
    ```

3.3.1.2 Transferring library members

FT systems transfer precisely one library member with an FT request. In the case of transfers between library members, the library organization, the access method and the record structure are retained, whereas in the event of transfers between library members and SAM files only the record structure is preserved.

– During file transfer with FTAM partners, library elements that are located in the remote system can only be accessed using a C-string (and not by using *LIB-ELEM()).

– Library members of a specific type (e.g. load modules) cannot be transferred unless the FT product in use at the partner system permits the transfer of members of this type.

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 76.

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.

The following files cannot be transferred by openFT:

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

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 77.

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.

During file transfer with FTAM partners, there is no blank line expansion. Tabulators are expanded during transfers using the character set *Graphic String,* but not in the *General String*. For more detailed information on FTAM character sets, see also section “FTAM files” on page 71.

Binary format

When “Binary format” is specified, it is assumed that the file to be transferred contains an unstructured sequence of binary data. In the receiving system, a file with an undefined record length is generated. The binary data remains the same.

User format

When sending a file, it is assumed that length fields divide up the file into records. The first two bytes of each record must indicate its length, including the length of the record length field. When the file is fetched, this length data is generated in accordance with the actual record lengths in the remote system. The contents of the records are treated like binary data, i.e. not converted.

Both the record structure and the binary data remain unchanged when a file is transferred. The record length fields are stored in all Unix and Windows systems starting with the most significant byte. The maximum permitted record length within a file in the user format depends on the operating system.

3.3.4 FTAM files

You can exchange the so-called “document types” FTAM-1 (for text files) and FTAM-3 (for binary files) with FTAM partners.

The file structure and contents of these FTAM files are described in the Kernel group in “contents-type”:

- constraint set
  The constraint set describes the file structure. The subset of the FTAM standard selected by the functional standard ISO/EN 10607-3 permits only the value *unstructured*. The *constraint set* also specifies the actions which are permissible with the file on the basis of the structure of the file. For unstructured files, read, overwrite, extend and delete operations are permitted. Together with the *permitted actions*, the *constraint set* restricts the set of possible actions on a file.

  document type
  describes the actual contents of the file. ISO/EN 10607-3 requires support of FTAM-1 (unstructured text) and FTAM-3 (unstructured binary) for files with binary contents. The string format (*string significance*) can be variable (*variable*), fixed (*fix*) or not significant for storage (*not significant*). Furthermore, a maximum length of the string (*maximum string length*) can also be defined.

  In the case of text files (FTAM-1), the *universal class number* specifies the characters present in the text:

  – *GraphicString* can contain all graphical character sets (G sets) and escape sequences can be used to switch between character sets (see ISO 2022).

    openFT sets the character set to ISO 646 IRV (or ASCII IRV or ISO 8859-1 G0 set) plus ISO 8859-1 G1 set which broadly covers the characters used in the European languages. When two partners interconnect with openFT as of V10, the character set for file transfer is set to UTF-8.

  – *GeneralString* may contain not only graphical characters but also control character sets (C sets) which can also be switched.

  – *VisibleString* contains only graphical characters from ISO 646 IRV.

  – *IA5String* contains graphical characters from ISO 646 IRV and control characters from ISO 646 (C0 set).

3.3.5 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. More details on file transfer with FTAM partners ca be found in the section “Special points for file transfer with FTAM partners” on page 106.

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).

Record structure in receive system	Local system: BS2000	Remote system	Direction ← / → [1]	DATA TYPE
system-conformant (in the usual manner in the receive system)	DMS, PLAM	BS2000: DMS, PLAM	← / →	N BCU
	DMS, PLAM, POSIX	BS2000: POSIX, Unix system, Windows, z/OS	← / →	N C

Explanation: N(ot specified), B(inary), C(haracter) and U(ser)

[1] ← = fetching, → = sending

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).

Record structure in receive system	**Local system: BS2000**	**Remote system**	**Direction ← / → [1]**	**DATA TYPE**
system-conformant (in the usual manner in the receive system)	DMS, PLAM	DMS, PLAM, z/OS	← / →	N BCU
	DMS, PLAM	POSIX	← / →	N C
	POSIX	POSIX	← / →	N BC
	POSIX	z/OS	←	C B
original record structure (usually in the send system)	POSIX	DMS, PLAM	→	B
	POSIX	Unix system, Windows	← / →	B
	DMS, PLAM	Unix system, Windows	←	B
user format (system-independent)	DMS, PLAM	POSIX	→	U
	DMS*, PLAM, POSIX	Unix system, Windows	→	U
	POSIX	Unix system, Windows	←	U
no record structure (i.e. the record structure may be lost)	DMS, PLAM	POSIX, Unix system, Windows, VMS	→	B

Explanation: N(ot specified), B(inary), C(haracter) and U(ser)
* applies only for SAM files with variable record length (RECFORM=V)

[1] ← = fetching, → = sending

Temporary files cannot be transferred with openFT.

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

– in transparent format, see page 74

– by specifying the target format, see the section “Heterogeneous transfer of PAM and ISAM files” on page 75

Record by record transfer

When DMS files are transferred between BS2000 systems, the file structure is not usually considered (the files are transferred block-by-block). The file's record structure is significant in the following cases (the files are transferred record-by-record):

– transfer between BS2000 and Unix systems, Windows or z/OS
– extension of a file with a record structure
– transfer of POSIX files
– transfer of library members

In these cases, the maximum length of the records that are to be transferred should not exceed the following values:

- Partner systems with openFT V10 and higher:
  - 32768 bytes in files with fixed-length records
  - 32760 bytes in files with variable-length records
- Partner systems with openFT < V10:
  - 32760 bytes in files with fixed-length records
  - 32758 bytes in files with variable-length records
  - 32248 bytes for compressed transfer (COMPRESS = *BYTE)

If files are transferred from Unix systems, Windows or POSIX to a DMS file then the maximum record length without the specification of the RECORD-SIZE parameter in the command may not exceed (2048*b-n) bytes, where b is the blocking factor of the BS2000 receive file (default on NK-PVS disks is b=2, otherwise the default is b=1). On K-PVS (K for Key) n=8, and on NK-PVS (NK for Non Key) n=20.

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.

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. The file content is transferred without the end-of-file marker of the C runtime system and empty blocks are initialized.
  The content of the PAMKEY and the architecture of the pubset are not transferred to the target system.
- 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. All blocks apart from the final one are completely filled with data and the end-of-file marker of the C runtime system is appended at the end.

  – If you transfer PAM files out of BS2000 and fetch them back again, the original pubset and the target pubset should, wherever possible, have the same block properties. If this is not the case, or if the content of the PAM keys is important for the file, you have to assume that the target file will become unusable.
  In the case of PLAM libraries, it is generally possible to retain the format during transfer. If the target pubset is NK4, the PLAM library must be converted with PAMCONV.

  – If a PAM file that has been swapped out is to be transferred back to a BS2000 system with openFT V10, it is stored as a sequential file with an undefined record format. The request is rejected under openFT with a Version < V10.

  – It is not possible to extend a PAM file by fetching a file from the foreign system.
- 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.

3.3.6 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 section “XHCS support by openFT” on page 80.

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 78.

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 local or 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. In the case of transfers with openFT partners on Unix or Windows systems, text files are recoded in the partner system.

A different mechanism is used between FTAM partners (see section “Virtual filestore” on page 106).

3.4.4 XHCS support by openFT

With XHCS, various codes can be used in a BS2000/OSD system at the same time. openFT can utilize XHCS information to recognize the code being used. Depending on the type and scope of the information, openFT (BS2000) employs the XHCS conversion tables (either before or after a file is transferred as a text file) to convert a file to a code that can be processed in the target system.

3.4.4.1 Binary file transfer

In the case of binary transfer, neither openFT for BS2000 nor the partner system converts the data to be sent or received to a different code. The data is stored by the receiving system in the form in which it was sent by the sending system. The user is responsible for checking that the receiving system supports the code used in the file.

If a CCSN is assigned as an attribute to the send file, openFT transfers the CCSN to the receiving system (openFT partners).

If the receiving system is openFT for BS2000/OSD, the CCSN of the send file is assigned as an attribute to the receive file without any further verification. Assignment is implemented even if the CCSN is not defined in the local system, i.e. no code tables with the name in question exist. It is not absolutely essential for XHCS to be loaded.

If the receiving system is not an openFT for BS2000/OSD, the CCSN is ignored.

If openFT receives a file from an openFT system that is not an openFT for BS2000/OSD, it does not receive information on the character set used. The user is responsible for assigning the CCSN to the file as an attribute, if the character set is defined in the local BS2000 system.

3.4.4.2 File transferred as text file

This section describes the scenario in which no CCS has been assigned to the text file in the transfer request. In other cases, the description given in the section “Specifying the CCS on a transfer request” on page 78 applies.

When transferring a file as a text file, openFT implements code conversion in accordance with the type of partner system and the coding information available to openFT.

The local system described below is a BS2000/OSD system with XHCS support. Various different scenarios must be distinguished (send and receive, any file attributes and the type of the partner system).

Sending

If a CCSN is assigned as an attribute to the send file in the local system, openFT converts the file to a reference code compatible to the character set of the file. openFT transfers the code-converted file, the CCSN of the file and the CCSN of the reference code to the partner system.

Example

A file stored in the local computer is in EBCDIC.EHC.LC code. The CCSN of the character set is EEHCLC. The file has the attribute CCSN=EEHCLC. The code EBCDIC.EHC.LC is compatible with the reference code EBCDIC.DF.04-5. The CCSN of the reference code is EDF045.

Prior to transfer, openFT converts the file into EBCDIC.DF.04-5.

openFT transfers the following to the partner system:
- the converted file
- the CCSN of the reference code (EDF045)
- the CCSN of the code of the original file on the local computer (EEHCLC)

If no CCS name has been assigned to the send file in the local system, openFT assumes that the file is coded in the unexpanded character set EBCDIC.DF.03.IRV. openFT no longer re-codes the send file before the file transfer. No CCSN is passed to the partner system.

openFT behaves in exactly the same way if the file has been assigned a CCSN. Moreover, there are no code tables in XHCS.

If the send file's CCSN corresponds to a 7-bit code table, the file will likewise be converted to the 8-bit reference code before it is sent.

If the code used in the file is not EBCDIC.DF.03.IRV, you should assign the CCSN of the code used to the send file prior to transfer or a CCSN is specified in the transfer request, so that openFT can convert the data accordingly prior to file transfer and send the requisite information to the partner system.

Example

The sending user ID on your BS2000 system has EBCDIC.EHC.LC assigned to it as its user default character set. The file to be transferred is also coded in the user default character set, but does not have a CCSN assigned to it as an attribute.
Under these circumstances, openFT transfers the file without first converting it to a reference code. The partner system receives no information concerning data code. Consequently, the partner system assumes that the receive file is coded in EBCDIC.DF.03.IRV. There is therefore a danger of data being corrupted in the receive file.

BS2000 provides a REP solution for this scenario in which files without a CCSN take the CCSN either from the user ID or from the host code.

The partner system interprets the information it receives on the character set according to its abilities. For example, openFT for Unix systems converts the data into the ISO 8859-n code with the variant number of the reference code.

Receiving

The receiving system must be able to distinguish the type of sending system, because the scope of information on the data code sent with the transferred file differs.

- The sending system is also an **openFT for BS2000/OSD with XHCS**

  The receiving openFT interprets the information on the CCSN and the reference code. If the receive file is to be created or an existing file overwritten, one of three possible cases may arise:

  a) The CCSN transferred with the send file is matched by a character set defined in the receive system and this character set is compatible with the reference code used for the file transfer.

     In this case the receiving openFT converts the received data into the corresponding character set by employing the tables that belong to the CCSN. The CCSN is included in the file's catalog entry as an attribute.

  b) In the receiving system, the CCSN transferred with the file is not matched by a character set compatible with the reference code used for file transfer. A user default character set has, however, been defined for the receiving user ID and this character set is compatible with the reference code used.

     In this case the local openFT converts the receive file into the user default character set. The CCSN of the user default character set is included in the file's catalog entry.

  c) In the receiving system, the CCSN transferred with the file is not matched by a character set compatible with the reference code used for file transfer. Moreover, no user default character set has been defined for the receiving user ID or the user default character set defined is not compatible with the reference code used for the file transfer.

     In this case, openFT does not implement conversion. The file is stored in the reference code of the send file and the CCSN of the reference code is included in the catalog entry of the receive file.

Example

The remote openFT sends a file to the local openFT instance. The reference code used for the file is EBCDIC.DF.04-2. In addition the file as such, the local openFT receives from the remote partner the CCSN of the reference code (EDF042) and the CCSN of the code in which the send file was coded on the remote BS2000 system (CCSN=EEHCL2 for EBCDIC.EHC.L2) before the remote openFT converted the file into the reference code.

In the local BS2000 system there is a character set with the CCSN EEHCL2. This character set is compatible with reference code EBCDIC.DF.04-2, which is the code of the received data.

The local openFT instance converts the incoming file into the corresponding code and assigns the CCSN EEHCL2 as an attribute to the file.

- The sending system is **not** an **openFT for BS2000/OSD**

  In this case the file transferred by the non-openFT system is coded in a reference code. The non-openFT system informs openFT of the name of this reference code. openFT is the receiving system and as such, it checks whether a user default character set is defined for the receiving user ID in the local system.

  If a user default character set exists and if it is compatible with the reference code transferred, the received file is converted into the user default character set and the corresponding CCSN is included in the file's catalog entry.

  If no user default character set has been defined or if it is incompatible with the reference code of the file, code conversion does not take place. Instead, the file is stored in the reference code of the send file. The CCSN of the reference code is assigned to the receive file as a file attribute.

  Example

  The receive file is coded in EBCDIC.DF.04-2. The user ID involved locally in file transfer has the user default character set EBCDIC.EHC.L2 (with CCSN EEHCL2). This user default character set is compatible with EBCDIC.DF.04-2.
  In this case, the local openFT converts the receive file into EBCDIC.EHC.L2. CCSN=EEHCL2 is assigned to the file as an attribute.

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 or processor name, see page 85) 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 87. 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.

ftam FTAM partner, i.e. communication takes place over the FTAM 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, only for FTP partners

– BCAM processor name, length 1 to 8 characters

– IPv4 address (only with FTP partners) 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.

– IPv6 address (only with FTP partners) with the prefix %ip6, i.e. for example `%ip6[FEDC:BA98:7654:3210:FEDC:BA98:7654:3210]` (ipv6) or `%ip6[FE80::20C:29ff:fe22:b670%5]` (ipv6 with scope ID)

The square brackets [..] must be specified.

The scope ID designates the local network card via which the remote partner can be accessed in the same LAN segment. It must be appended to the address with a % character. In Windows systems, this is a numerical value (e.g. 5). On other systems, it may also be a symbolic name (e.g. *eth0*). The scope ID can be identified using the *ipconfig* command.

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;

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

4800 for FTAM partners

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**

– FTAM partner:
Length 1 to 10 characters; a printable selector will be coded as variable length ASCII in the protocol. Exception: T-selectors that start with $FTAM (default value) are coded in EBCDIC and padded with spaces to the length of 8 characters.

All alphanumeric characters and the special characters @ $ # _ - + = and * can be used with ASCII selectors.

Default value: **$FTAM**

Note:
– As a rule, **SNI-FTAM** must be specified for Windows partners with openFT-FTAM up to V10. As of openFT-FTAM V11 for Windows, the default value has been changed to **$FTAM** and can therefore be omitted.
– In openFT, printable transport selectors are always used with uppercase characters even if they are specified or output in lowercase characters.

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.
Presentation 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 interpreted as variable length ASCII in the protocol.

Default value: empty

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.

- FTAM partner systems on which no file transfer product of the openFT product family is used:

  `user-identity,[storage account],filestore-password`

- 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 and access modalities for FTAM partners.

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.

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.

  When a file transfer request is accepted, a lock is set on each file to be transferred. Only read access is granted to other users for the send files; no access is permitted for the receive files. The BS2000 command SHOW-FILE-LOCK indicates whether a file has been locked by openFT and lists the transfer ID's involved. File locks are automatically removed on unloading the subsystem.

  The mode of operation of openFT dictates that a receive file which already exists can only be overwritten if both read and write access are available for this file. For file access, the specifications of the ACL (Access Control List) and BASIC-ACL must be adhered to.

- 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.

- 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.

The following table shows the conditions under which the FT user can access a BS2000 file:

Access mode	Conditions for file access
Read of a sending file	– File cataloged under specified login name or – File defined as multiuser or – User is working under the login name TSOS and – Write access is permitted and – Valid password was given, when the file to be transferred is read and execute protected by a password
Overwrite an existing receive file	– File cataloged under specified login name or – File defined as multiuser or – User is working under the login name TSOS and – Read and write access is permitted and – Valid password was given, when the file to be transferred is read-protected by a password

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.

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.

Data compression is not supported on links to FTAM partners.

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 48).

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. No data encrypted by openFT can be exchanged with FTAM 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 Transfer of protection attributes

If the partner is a BS2000 system running openFT as of V11.0, DMS files can be transferred in such a way that the protection attributes USER-ACCESS, ACCESS, BASIC-ACL, EXPIRATION-DATE, FREE-FOR-DELETION and DESTROY are also transferred in addition to the default file attributes.

The protection attributes can only be transferred via the openFT protocol and they cannot be transferred in transparent mode. The target file must be created as new or overwritten and must not be a file generation.

3.6.6 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.

3.6.7 Access mode

It is possible to define FTAM-specific file attributes for file transfer with FTAM. The FTAM file attributes that describe the file type must be identical to those specified in the file transfer request. The corresponding attributes are presented in the section “Mapping FTAM attributes to the real file system” on page 109ff.

3.6.8 Preprocessing and postprocessing

The “preprocessing” and “postprocessing” functions make it possible to execute any 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 98.

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.

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 pre- or postprocessing are running in BS2000, the commands will be executed as a batch job:

- During preprocessing, data output to SYSLST or the %TEMPFILE variable must occur. If you specify %TEMPFILE then the file can have a file format other than SAM-V. The batch job uses this output to create a help file, which is then transferred to the partner system.Because the file attributes of the generated file are not known when the request is accepted, they must either be passed as request parameters (RECORD-SIZE, RECORD-FORMAT) or the transfer must be performed transparently if the systems are homogeneous.
- During postprocessing, the data is read from SYSDTA by default. In this event, you must have a format which can be processed by SYSDTA (SAM-V or ISAM-V). It is also possible, however, to explicitly address the transferred file using the variable %TEMPFILE. For this, the file can be of a different file format than SAM-V or ISAM-V. openFT makes the transferred data available to the first command and waits until processing is complete ("postprocessing").

Each individual BS2000 command must be preceded by a forward slash '/'.

Preprocessing and postprocessing in the BS2000 system runs under a separate TSN. If an ENTER-JOB command is encountered during preprocessing, this command is subject to the usual definitions for the user ID that initiated preprocessing. openFT generates a JOB file with the name $*userid*.S.VV.FT.*unique-string*.JOB which is automatically deleted after the ENTER-JOB. If a job class JBCLJOB has been declared for the user under whose ID the pre/postprocessing is being performed, then pre/postprocessing is performed under this job class. Otherwise the user's default job class is used.

If the %JOBCLASS variable is used, preprocessing and postprocessing can be started under a selected job class by inserting the following command in the file pre/postprocessing operation:

```
/REMARK %JOBCLASS=<jobclass>
```

When choosing the job class, you should note that pre/postprocessing operations should be started as quickly as possible as otherwise resources (transport connections, server tasks) may block and, in the worst scenario, may be aborted after 15 minutes.

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. 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 SYSLSTor %TEMPFILE output is transferred. The SYSLST or %TEMPFILE output is temporarily stored in a file prefixed by S.PP 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 automatically deleted as soon as transfer and/or preprocessing and postprocessing are completed.

  If the FT subsystem is closed down before a file can be deleted, then the information concerning the file for deletion is lost and automatic deletion is no longer possible.
- 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).

The openFT subsystem cannot be stopped, as long as there are still restartable requests, whose pre- and postprocessing sessions have not been completed. If this is taking too long, the system administrator can cancel the batch jobs belonging to the requests (using the CANCEL command). In this case, the message FTR2083 is displayed during the next restart of openFT.

If restartable requests are still active in the command execution phase when openFT is stopped using STOP-FT, then shut-down is delayed by up to 2 minutes. If command execution still has not been completed after 2 minutes, openFT will be stopped and the request will be cancelled during the next START-FT.

Server function for remote command execution

One special form of preprocessing is the server function for the remote command execution (EXECUTE-REMOTE-CMD 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. EXECUTE-REMOTE-CMD 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 BS2000 commands, then the command's SYSLST output is routed to stdout and the SYSOUT output to stderr.

In connection with the subject of preprocessing, please also see the examples in the description of the TRANSFER-FILE command on page 403ff.

3.6.9 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 85. In addition, all characters apart from letters, digits or periods are replaced by '@'.

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 or FTAM partners, since the FTP or FTAM protocol does not permit transfer of follow-up processing data. Follow-up processing in the FTP or FTAM 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 (not for FTAM partners), 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.

3.6.10 Monitoring using job variables

openFT makes it possible to use a job variable to monitor an FT request. The name of the job variable together with any password that may be necessary in order to access this job variable are specified in the FT request. If the job variable does not yet exist, it is entered in the catalog. An existing job variable is overwritten unless it is currently monitoring another FT request. The user must also be authorized to access this job variable.

A job variable that is monitoring an FT request is not protected against write accesses. Processing of the FT request continues even if openFT recognizes that the content of a job variable is inconsistent. However, in this case the job variable is no longer used. The system issues the following warning at the terminal at which the command was issued:

```
FTR0802 Request (&00). Warning: content of jobvariable inconsistent
```

This warning also appears in the result list if one is generated.

A job variable that monitors a request is also used to identify the request and can be used as a selection criterion (to cancel the request in the CANCEL-FILE-TRANSFER command (CNFT / NCANCEL), for information about the command in the SHOW-FILE-TRANSFER command (SHFT / NSTATUS). It can also be used for event control purposes.

Contents of the job variable

Column	Length	Meaning
1 - 2	(2)	Transfer status with the possible values:
		'$W' The FT request has been accepted and is waiting for the necessary resources or for restart. '$R' The FT request is running. '$A' The FT request has terminated incorrectly or has been deleted by means of a command. '$T' The FT request has been terminated successfully.
3 - 3	(1)	Space
4 - 14	(11)	Transfer identification
15 - 15	(1)	Space
16 - 71	(56)	Local file or library name
72 - 72	(1)	Space
73 - 136	(64)	Local member name when library members are transferred, otherwise space
137 - 137	(1)	Space
138 - 145	(8)	Local member type when library members are transferred, otherwise space

Column	Length	Meaning
146 - 146	(1)	Space
147 - 153	(7)	Message number for status '$T' or '$A', otherwise space
154 - 154	(1)	Space
155 - 218	(64)	Additional information about the termination of a request
219 - 247	(29)	Space
248 - 256	(9)	'FT–BS2000' or space. While the job variable is monitoring an FT request, this area contains the entry 'FT–BS2000'. Any other FT requests that want to use this variable are rejected. When transfer is complete, the field is overwritten with spaces.

3.7 File management

File management in openFT is possible both in the remote and in the local system.

3.7.1 File management in the remote system

openFT offers the option of managing remote system files from the local system (file management). In the partner system, you can

– list the contents of directories,
– query file attributes, e.g. query the size of a send file,
– modify file attributes, e.g. rename files,
– delete files.
– create, rename and delete directories

openFT for Windows and openFT for Unix systems also offer the option of renaming, creating or deleting directories in openFT partner systems. Partner systems, which support the file management function can also assume the initiative for such requests and access their local system accordingly from the remote system. In both cases, the system in which the initiative has been taken sends a description of the request to the partner system. The partner system executes the request according to its conventions.

If the partner system is a z/OS system, a number of special issues need to be observed. You will find details in the User Guide for openFT for z/OS.

The file management functions are performed via the appropriate protocols (openFT, FTAM or FTP). You can detect differences in the protocols between openFT, FTAM and FTP partner systems by changing the file attributes. Depending on the protocol, and what the partner system supports, you can modify the following attributes of a file.

Attribute	**FTAM partner**	**openFT partner**	**FTP partner**
File name (FILE-NAME/NEW-NAME)	X	X	X
Access rights (ACCESS-MODE)	X	X	
File availability (FILE-AVAILABILITY)	X		
Account for file storage costs (STORAGE-ACCOUNT)	X		
Legal qualification for using a file (LEGAL-QUALIFICATION)	X		
Future file size (FUTURE-FILE-SIZE)	X		

3.7.2 File management in the local system

When using the FTAM functionality, you have the option of assigning special FTAM attributes to file in the local system for communication with FTAM partners (see page 106). The functionality offered by this approach allows you to display and modify FTAM attributes of a file in the local system.

The FTAM attributes exist only in the virtual filestore and primarily valid for file transfer and file management with FTAM partners. In the local system, the operating-system specific setting of the file attributes remains unaltered. This means that This means that files and file attributes can still be modified using commands specific to the operating system. For example, a file can be deleted using a system-specific delete command although the corresponding setting of PERMITTED-ACTION prohibits deletion of the file for FTAM partners.

The following table shows the file management functions in the local system:

FTAM attribute	**display**[1]	**modify**
FILE-NAME *	X	
STORAGE-ACCOUNT	X	
Type of last file usage *	X	
Name of last user of file *	X	
Date and time of last change of file contents	X	
DATA-TYPE	X	X
CHARACTER-SET *	X	X
RECORD-FORMAT *	X	X
Maximum record length (RECORD-SIZE) *	X	X
File availability (FILE-AVAIAIBILITY) *	X	
Access rights (PERMITTED-ACTIONS) *	X	X
Current file size in bytes (CURRENT-FILE-SIZE) *	X	
Possible file size in bytes (FUTURE-FILE-SIZE)	X	
Legal qualifications (LEGAL-QUALIFICATION)	X	

[1] Only the FTAM attributes marked with * are displayed for local file management; all attributes are displayed for remote file management.

The following FTAM attributes are evaluated for file transfers using the openFT protocol and in part for the FTP protocol:

- Data type (DATA-TYPE)
- Record format (RECORD-FORMAT)
- Maximum record length (RECORD-SIZE)

If the format attributes specified in the file transfer request are not consistent with these FTAM attributes, the request is generally rejected. To avoid this, the FTAM attributes can be deleted in the local file without deleting the file itself.

However, these FTAM attributes are only set for file transfer requests using the FTAM protocol (not for requests via the openFT or FTP protocol).

3.8 Special points for file transfer with FTAM partners

The FTAM functionality allows you to execute file transfer on the basis of ISO protocol ISO8571. The sections below describe special points for “FTAM specialists” with respect to transfer and mapping of FTAM-specific file attributes for file transfer with FTAM partners.

3.8.1 Virtual filestore

Any system that is to enable file transfer using FTAM protocols must make its files available to partner systems in a format that is defined by standard (ISO8571). For this purpose a file’s attributes are mapped from the real filestore onto a virtual filestore and vice versa. The virtual filestore thus has no effect on the attributes of the files in the local system, but has only the tasks of transporting file attributes to the remote FTAM system. In the sections below, the criteria for describing a a file in the virtual filestore are introduced. The format of the virtual filestore is defined by the FTAM standard. Basically, a distinction is made between three different groups of file attributes:

Kernel group

describes the basic attributes of the files. These are specified when the file is created. They include the file name, information relating to the file structure and file contents, and details of agreed file access rights.

Storage group

covers the storage attributes of files. The storage attributes include the file size, the file availability, the date and time of the last read or write access, as well as identification of the user who initiated this in access.

Security group

defines the security attributes for access protection.

Attributes of the kernel group

The attributes in the kernel group are set when the file is created, and contain the basic information on a file:

file name
: contains the file name.

permitted actions
: define which actions can be performed for a certain file:
  – read file (READ-FILE)
  – insert data unit (INSERT-DATA-UNIT)
  – replace (overwrite) file (REPLACE-FILE)
  – extend file (EXTEND-FILE)
  – erase data unit (ERASE-DATA-UNIT)
  – read file attributes (READ-ATTRIBUTES)
  – modify file attributes (CHANGE-ATTRIBUTES)
  – delete file (DELETE-FILE)

  The *permitted actions* also define the method that can be used to access structured files (see also the section “FTAM files” on page 71).

  – forwards (TRAVERSAL)
  – backwards (REVERSE TRAVERSAL)
  – any (RANDOM)

contents type
: Defines the data structure and the method that can be used to access the structured data.

Attributes of the storage group

The attributes of the storage group describe the filestore properties, for example who last accessed the file, the type of access, and when. Some of these properties are automatically modified when the file is read or modified. However, they cannot be modified directly using user commands. You can influence directly modifiable attributes with openFT.

Attribute[1]	**Definition**
storage account *	identifies who is responsible for the file storage costs
date and time of creation	indicates the date and time of creation
date and time of last modification	indicates the date and time of the last modification
date and time of last read access	indicates the date and time of the last read access
date and time of last attribute modification	indicates the date and time of the last attribute modification

Attribute[1]	Definition
identity of creator	identifies the user who created the file
identity of last modifier	identifies the user who last modified the file
identity of last reader	identifies the user who last read the file
identity of last attribute modifier	identifies the user who last modified the file attributes
file availability *	provides information on whether a file is available immediately, or whether it must first be obtained, e.g. from an archive
filesize	describes the storage capacity occupied in the actual filestore. A file can thus differ in size in systems that display file types in different ways. Some filestores assign a multiple of a basic unit, e.g. blocks, for file storage. *file size* thus specifies a value that does not correspond to the file size
future filesize *	describes the future file size, i.e. possible file size after processing. The initiator can modify the *future file size* value. As soon as the file reaches the specified file size, the responder can increase the value with or without a warning to the initiator. Alternatively, the responder can reject the modification of a value with an appropriate error message.

[1] Attributes marked with * can be modified directly.

Attributes of the security group

The FTAM virtual filestore concept provides a security group for access protection.

Attribute[1]	Definition
access control *	indicates the conditions governing access to files. For example, this may include passwords for various types of access (read, insert, replace, extend), or locks that are used to regulate simultaneous access to a file by different users.
legal qualifications *	specify the legal status of the file and its usage. At present, there is no accepted interpretation of this attribute, i.e. its interpretation depends on the particular partner.

[1] Attributes marked with * can be modified directly.

3.8.2 Mapping FTAM attributes to the real file system

This section describes the way in which the FTAM implements the virtual filestore, and the mechanisms used for mapping virtual and real filestores in BS2000/OSD.

Some FTAM attributes are mapped to the attributes available in BS2000/OSD, and others to the so-called “FTAM catalog”. Attributes that have no significance in BS2000 are ignored.

The FTAM catalog is used to extend the file attributes available in BS2000. It is only relevant for access using FTAM. This means that a file can be deleted using a BS2000 command (e.g. DELETE-FILE) , even if the *permitted actions* parameter from the FTAM catalog does not permit this for an FTAM partner.

Entries in the FTAM catalog are created using inbound file management requests or a file transfer request, or by modifying the local FTAM attributes. When the file is deleted from the remote system, the appropriate entry in the FTAM catalog is also removed.

It is important to remember that a file identified as a text file in the FTAM catalog, for example, cannot be transferred as a binary file, nor can it be extended by binary data.

3.8.2.1 Inbound mapping of FTAM attributes

The following table shows how FTAM attributes are mapped to the real BS2000 file system.

Attribute group	FTAM attributes	Mapping in BS2000 (inbound receive)	Modify FTAM attributes
Kernel group	permitted actions READ-FILE INSERT-DATA-UNIT REPLACE-FILE EXTEND-FILE ERASE-DATA-UNIT READ-ATTRIBUTES CHANGE-ATTRIBUTES DELETE-FILE	FTAM catalog	permitted locally 1)
	universal class number GRAPHIC GENERAL IA5 VISIBLE	FTAM catalog	permitted locally 1)
	string significance UNCHANGED VARIABLE FIXED NOT-SIGNIFICANT	FTAM catalog	permitted locally 1)
	maximum string length	FTAM catalog	permitted locally 1)

Attribute group	FTAM attributes	Mapping in BS2000 (inbound receive)		Modify FTAM attributes
	document type FTAM1 FTAM3	FTAM catalog		permitted locally 1)
Storage group	file availability IMMEDIATE DEFERRED future file size storage account	FTAM catalog ignored ignored		inbound permitted for DMS files, not permitted for POSIX files and lib members not permitted not permitted
Security group	ActionList (of 1ACE)	with (BASIC)ACL	without (BASIC)ACL	
	READ-FILE INSERT-DATA-UNIT REPLACE-FILE EXTEND-FILE ERASE-DATA-UNIT READ-ATTRIBUTES CHANGE-ATTRIBUTES DELETE-FILE	(BASIC-) ACL=r not permitted w w w 3) 4) w	ACCESS=READ not permitted WRITE WRITE WRITE READ READ WRITE	inbnd. permitted 5) not permitted inbnd. permitted 5) inbnd. permitted 5) inbnd. permitted 5) not permitted 2) not permitted 2) inbnd. permitted 5)
	LEGAL-QUALIFICATION	ignored		not permitted

1) The local FTAM attribute can be modified locally using the command MODIFY-FILE-FT-ATTRIBUTES.

2) The value must always be sent but can never be modified.

3) If group or other have no rights due to (BASIC)-ACL, the request would be rejected if a user ID belonging exclusively to group or other was specified.

4) File attributes can only be modified by the owner. In other words, any user ID that is not the owner and which belongs to group or other cannot modify file attributes.

5) ACL attributes cannot be modified.

The following file attributes are derived from the current BS2000 file attributes:

– file name
– file size
– future filesize
– identity of creator (always identical to owner ID)
– date and time of last read access (corresponds to LADATE and LATIME)
– date and time of last attribute modification (corresponds to CRDATE/CRTIME)
– date and time of last modification (corresponds to CRDATE/CRTIME)
– access control
– string significance

Other attributes are only partially supported by openFT for BS2000. As the responder, openFT for BS2000 does not return any value for the following file attributes (*no value available*):

– identity of last modifier
– identity of last reader
– identity of last attribute modifier
– storage account
– legal qualification

In BS2000, the FTAM protocol parameter *filestore password* is mapped to the password of the of the login name concerned.

3.8.2.2 Inbound mapping the document type

If the initiative lies with the FTAM partner, then only SAM files and library elements can be transferred (in other words, no ISAM or PAM files). In the process, openFT for BS2000 maps the file formats to the virtual filestore of the partner system.

In contrast, if files are transferred from FTAM partners to openFT for BS2000, the file format from the virtual store is mapped to a SAM file format. If files are transferred from FTAM partners to openFT for BS2000 and are to be stored as library members, there is no “memory” for the FTAM-specific attributes. Therefore, in this case there is no FTAM catalog entry.

The following table provides details on mapping FTAM attributes to the real BS2000 file system and vice versa.

openFT as the responder and receive system (FTAM → BS2000 receive files)

FTAM (virtual filestore of the remote system)			BS2000 file	
document type	universal class	string significance		
FTAM-1	25 - GraphicString	variable	SAM	V
FTAM-1	26 - VisibleString	variable	SAM	V
FTAM-1	27 - GeneralString	not-significant	SAM	V
FTAM-1	22 - IA5String	not-significant	SAM	V
FTAM-1	25 - GraphicString	fix	SAM	F
FTAM-1	26 - VisibleString	fix	SAM	F
FTAM-3	————	fix	SAM	F
FTAM-3	————	variable	SAM	V
FTAM-3[1]	————	not significant	SAM	U

[1] SAM files that were not created/processed via FTAM facilities have the specified structure attribute combinations (depending on the SAM record format) if the initiator does not specify different structure attributes.

If FTAM-1 files are mapped to the BS2000 real storage, the data is converted to the EBCDIC.DF.04 format and transferred as text files. FTAM-3 files are not converted, in other words they are transferred as binary files.

If a partner specifies file format attributes while attempting to write or extend files in BS2000 without regenerating them, the attributes specified must be compatible with the attributes of the existing file. This means:

- The *document type* which the initiator may have specified must correspond to the data type (DATA-TYPE) of the file.

document type	DATA-TYPE
FTAM-1 (unstructured text)	*CHARACTER or -
FTAM-3 (unstructured binary)	*BINARY or -
not specified	any

- If the initiator specifies the *document type* parameter *string significance*, it must correspond to the record format (RECORD-FORMAT) of the file as displayed in the SHOW-FILE-FT-ATTRIBUTES command.

string significance	**RECORD-FORMAT**
variable	v
fixed	f
not significant	u
not specified	any

- The following combinations are possible for the *character repertoire* and the *universal class number* (*document type* parameter, only with *unstructured text*):

universal class number	**character repertoire**
GRAPHIC	*GRAPHIC
VISIBLE	*GRAPHIC or *VISIBLE
GENERAL	*GENERAL
IA5	*GENERAL or *IA5

openFT as the responder and send system (FTAM → BS2000 send file)

FTAM (virtual filestore of the remote system)			**BS2000 file**	
document type	universal class	string significance		
FTAM-1	25 - GraphicString	variable	SAM	V
FTAM-1	26 - VisibleString	variable	SAM	V
FTAM-1	27 - GeneralString	not significant	SAM	V
FTAM-1	22 - IA5String	not significant	SAM	V
FTAM-1	25 - GraphicString	fix	SAM	V
FTAM-1	26 - VisibleString	fix	SAM	F
FTAM-3	———	fix	SAM	F
FTAM-3	———	variable	SAM	V
FTAM-3	———	not significant	SAM	U

Files that are mapped to FTAM-1 files must be in EBCDIC.DF.04 format (see code tables). Code extensions by means of escape sequences are not permitted. FTAM-3 files are not converted.

If a partner attempts to read files in BS2000 and specifies the file format attributes, the specifications must be compatible with those of the existing file. This means:

- The *document type*, which the initiator may have specified must correspond to the data type (DATA-TYPE) of the file.

document type	**DATA-TYPE**
FTAM-1 (unstructured text)	*CHARACTER or -
FTAM-3 (unstructured binary)	*BINARY or -
not specified	any

- If the initiator specifies the *document type* parameter *string significance*, it must correspond to the record format (RECORD-FORMAT) of the file as displayed in the SHOW-FILE-FT-ATTRIBUTES command.

string significance	**RECORD-FORMAT**
variable	v
fixed	f
not significant	u
not specified	any

- The following combinations are possible for the *character repertoire* and the *universal class number* (*document type* parameter, only with *unstructured text*):

universal class number	**character repertoire**
GRAPHIC	*VISIBLE or *GRAPHIC
VISIBLE	*VISIBLE
GENERAL	*GENERAL or *IA5
IA5	*IA5

3.8.2.3 Access protection

As explained in the section “Virtual filestore” on page 106, openFT supports the security die Security group of the virtual filestore. This provides an effective protection mechanism against unauthorized access to files.

For access authorization to the virtual filestore of a system you need the FTAM protocol parameters *initiator identity* and *filestore password*. openFT for BS2000 maps these parameters to the BS2000 parameters USER-IDENTIFICATION (user ID) and PASSWORD (password of the user ID).

For file transfers with FTAM partners it is also possible to use the the functions of the add-on product openFT-AC for extended protection against unauthorized forms of access. If an admission profile in BS2000 is to be addressed by an FTAM partner, then the transfer admission (TRANSFER ADMISSION) for the profile concerned must be supplied in the protocol parameter *initiator identity*. The parameters *filestore password* and *account* must not be specified. Apart from this, the rules of the openFT-AC described in this manual apply here (e.g. referencing a file that has been predefined in the admission profile either with the specification *NOT-SPECIFIED for the file name).

3.8.2.4 Outbound mapping of the document type

If openFT for BS2000 is the initiator, the FT user can use the file type specification (DATA-TYPE= *CHARACTER/*BINARY/*NOT-SPECIFIED) to specify in the request whether text or binary data is to be transferred. There is no attribute for binary or text data in the real store on the BS2000/OSD.

The following tables provide information on mapping the *document type* during file transfer. A distinction is made here between openFT as the receiving system and the sending system.

openFT as initiator and send system (BS2000 send file → FTAM)

Case a1:
Transfer a text file to the FTAM partner. No entries in the local FTAM catalog,

Entries in the TRANSFER-FILE(NCOPY) command:

```
TRANSFER-DIRECTION = TO
DATA-TYPE = *CHARACTER
WRITE-MODE = any
```

BS2000		**FTAM (virtual filestore of the remote system)**		
send file		document type	universal class	string significance
SAM	V	FTAM-1	25 - GraphicString	variable
SAM	F	FTAM-1	25 - GraphicString	fix
SAM	U	not supported as FTAM text file		

Files that are mapped to FTAM-1 files must be in EBCDIC.DF.04 format (see code tables). Code extensions by means of escape sequences are not permitted.

Case a2:
Transfer a text file to the FTAM partner,

The entry in the local FTAM catalog is DATA-TYPE=*CHARACTER(*GRAPHIC) or DATA-TYPE=*CHARACTER(*VISIBLE)

BS2000		**FTAM (virtual filestore of the remote system)**		
send file		document type	universal class	string significance
SAM	V	FTAM-1	see FTAM catalog	variable
SAM	F	FTAM-1	see FTAM catalog	fix
SAM	U	not supported as FTAM text file		

Files that are mapped to FTAM-1 files must be in EBCDIC.DF.04 format (see code tables). Code extensions by means of escape sequences are not permitted.

Case a3:
Transfer a text file to the FTAM partner,

The entry in the local FTAM catalog is DATA-TYPE=*CHARACTER(*GENERAL) or DATA-TYPE=*CHARACTER(*IA5)

BS2000		**FTAM (virtual filestore of the remote system)**		
send file		document type	universal class	string significance
SAM	V	FTAM-1	see FTAM catalog	not-significant
SAM	F	not supported		
SAM	U	not supported		

Files that are mapped to FTAM-1 files must be in EBCDIC.DF.04 format (see code tables). Code extensions by means of escape sequences are not permitted.

Case b1:
Transfer a binary file to the FTAM partner,

No entry in the local FTAM catalog or the entry is DATA-TYPE=*BINARY

Entries in the TRANSFER-FILE(NCOPY) command:

```
TRANSFER-DIRECTION = TO
DATA-TYPE = *BINARY
WRITE-MODE = any
```

BS2000		**FTAM (virtual filestore of the remote system)**		
send file		document type	universal class	string significance
SAM	V	FTAM-3	————	variable
SAM	F	FTAM-3	————	fix
SAM	U	FTAM-3	————	not-significant

Transferring FTAM-3 files with variable *string significance* is not provided for in the function standard EN 10607-3. openFT provides additional support for this function because this file format corresponds to the user format in Unix and Windows systems.

FTAM-3 files are not converted.

Case b2:
Transfer a structured binary file with variable record length,

No entry in the local FTAM catalog or the entry is DATA-TYPE=*BINARY

Entries in the TRANSFER-FILE(NCOPY) command:

```
TRANSFER-DIRECTION = TO
DATA-TYPE = *USER
WRITE-MODE = any
```

BS2000		**FTAM (virtual filestore of the remote system)**		
send file		document type	universal class	string significance
SAM	V	FTAM-3	————	variable
SAM	F	illegal combination		
SAM	U	illegal combination		

Transferring FTAM-3 files with variable *string significance* is not provided for in the function standard EN 10607-3. openFT provides additional support for this function because this file format corresponds to the user format in Unix and Windows systems.

FTAM-3 files are not converted.

Case c1:
Transfer a file to the FTAM partner. No entry in the local FTAM catalog,

Entries in the TRANSFER-FILE(NCOPY) command:

```
TRANSFER-DIRECTION = TO
DATA-TYPE = *NOT-SPECIFIED
WRITE-MODE = any
```

BS2000		**FTAM (virtual filestore of the remote system)**		
send file		document type	universal class	string significance
SAM	V	FTAM-1	25 - GraphicString	variable
SAM	F	FTAM-1	25 - GraphicString	fix
SAM	U	FTAM-3	————	not-significant

Files that are mapped to FTAM-1 files must be in EBCDIC.DF.04 format (see code tables). Code extensions by means of escape sequences are not permitted. FTAM-3 files are not converted.

Case c2:
Transfer a file to the FTAM partner,

The entry in the local FTAM catalog is DATA-TYPE=*CHARACTER(*GRAPHIC) or DATA-TYPE=*CHARACTER(*VISIBLE)

BS2000		**FTAM (virtual filestore of the remote system)**		
Send file		document type	universal class	string significance
SAM	V	FTAM-1	see FTAM catalog	variable
SAM	F	FTAM-1	see FTAM catalog	fix
SAM	U	not supported as FTAM text file		

Files that are mapped to FTAM-1 files must be in EBCDIC.DF.04 format (see code tables). Code extensions by means of escape sequences are not permitted.

Case c3:
Transfer a file to the FTAM partner,

The entry in the local FTAM catalog is DATA-TYPE=*CHAR(*GENERAL). or DATA-TYPE=*CHARACTER(*IA5)

BS2000		**FTAM (virtual filestore of the remote system)**		
send file		document type	universal class	string significance
SAM	V	FTAM-1	see FTAM catalog	not-significant
SAM	F	not supported		
SAM	U	not supported		

Files that are mapped to FTAM-1 files must be in EBCDIC.DF.04 format (see code tables). Code extensions by means of escape sequences are not permitted.

Case c4:
Transfer a file to the FTAM partner,

The entry in the local FTAM catalog is DATA-TYPE=*BINARY

BS2000		**FTAM (virtual filestore of the remote system)**		
send file		document type	universal class	string significance
SAM	V	FTAM-3	————	variable
SAM	F	FTAM-3	————	fix
SAM	U	FTAM-3	————	not-significant

Transferring FTAM-3 files with variable *string significance* is not provided for in the function standard EN 10607-3. openFT provides additional support for this function because this file format corresponds to the user format in Unix and Windows.

FTAM-3 files are not converted.

Case d1:
Transfer a library member to the FTAM partner,

In this case, the send file is a library member.

BS2000	**FTAM (virtual filestore of the remote system)**		
send file (member type)	document type	universal class	string significance
S	FTAM-1	25 - GraphicString	variable
R	not transferable		
C, L	not transferable		
other	FTAM-1	25 - GraphicString	variable

Files that are mapped to FTAM-1 files must be in EBCDIC.DF.04 format (see code tables). Code extensions by means of escape sequences are not permitted.

Case d2:
Transfer a library member to the FTAM partner as a binary file,

In this case, the send file is a library member.

BS2000	**FTAM (virtual filestore of the remote system)**		
send file (member type)	document type	universal class	string significance
S	FTAM-3	————	variable
R	not transferable		
C, L	not transferable		
other	FTAM-3	————	variable

Transferring FTAM-3 files with variable *string significance* is not provided for in the function standard ENV10607-3. openFT provides additional support for this function because this file format corresponds to the user format in Unix and Windows systems.

openFT is initiator and receive system (FTAM → BS2000 receive file)

Case e:
Transfer a text file from the FTAM partner

Entries in the TRANSFER-FILE(NCOPY) command:

```
TRANSFER-DIRECTION = FROM
DATA-TYPE = *CHARACTER
WRITE-MODE = any
```

FTAM (virtual filestore of the remote system)			**BS2000 receive file**	
document type	universal class	string significance		
FTAM-1	25 - GraphicString	variable	SAM	V
FTAM-1	26 - VisibleString	variable	SAM	V
FTAM-1	27 - GeneralString	not-significant	SAM	V
FTAM-1	22 - IA5String	not-significant	SAM	V
FTAM-1	25 - GraphicString	fix	SAM	F
FTAM-1	26 - VisibleString	fix	SAM	F

If the BS2000 receive file already exists as a SAM file of a different record type and WRITE-MODE=EXTEND is specified in the FT request, the request is rejected.

Likewise, the request is rejected when WRITE-MODE=EXTEND if the information from the virtual filestore of the remote system is not compatible with the entries in the local FTAM catalog for the file that is to be extended.

If FTAM-1 files are mapped to the real filestore of BS2000, the data is converted to EBCDIC.DF.04 format (see code tables).

Case f1:
Transfer a binary file from the FTAM partner

Entries in the TRANSFER-FILE(NCOPY) command:

```
TRANSFER-DIRECTION = FROM
DATA-TYPE = *BINARY
WRITE-MODE = any
```

FTAM (virtual filestore of the remote system)			**BS2000 receive file**	
document type	universal class	string significance		
FTAM-3	————	fix	SAM	F
FTAM-3	————	variable	SAM	V
FTAM-3	————	not-significant	SAM	U

Transferring FTAM-3 files with variable *string significance* is not provided for in the function standard EN 10607-3. openFT provides additional support for this function because this file format corresponds to the user format in Unix and Windows systems.

If the BS2000 receive file already exists as a SAM file of a different record type and WRITE-MODE=EXTEND is specified in the FT request, the request is rejected.

Likewise, the request is rejected when WRITE-MODE=EXTEND if the information from the virtual filestore of the remote system is not compatible with the entries in the local FTAM catalog for the file that is to be extended.

FTAM-3 files are not converted.

Case f2:
Transfer a structured binary file with variable record length from the FTAM partner

Entries in the TRANSFER-FILE(NCOPY) command:

```
TRANSFER-DIRECTION = FROM
DATA-TYPE = *USER
WRITE-MODE = any
```

FTAM (virtual filestore of the remote system)			**BS2000 receive file**	
document type	universal class	string significance		
FTAM-3	————	variable	SAM	V

Case g:
Transfer a file from the FTAM partner without specifying data type

Entries in the TRANSFER-FILE(NCOPY) command:

```
TRANSFER-DIRECTION = FROM
DATA-TYPE = *NOT-SPECIFIED
WRITE-MODE = any
```

FTAM (virtual filestore of the remote system)			**BS2000 receive file**	
document type	universal class	string significance		
FTAM-1	25 - GraphicString	variable	SAM	V
FTAM-1	26 - VisibleString	variable	SAM	V
FTAM-1	27 - GeneralString	not-significant	SAM	V
FTAM-1	22 - IA5String	not-significant	SAM	V
FTAM-1	25 - GraphicString	fix	SAM	F
FTAM-1	26 - VisibleString	fix	SAM	F
FTAM-3	————	fix	SAM	F
FTAM-3	————	variable	SAM	V
FTAM-3	————	not-significant	SAM	U

Files with different combinations of the file characteristics in the virtual filestore cannot be transferred with openFT.

If the BS2000 receive file already exists as a SAM file of a different record type and WRITE-MODE=EXTEND is specified in the FT request, the request is rejected.

Likewise, the request is rejected when WRITE-MODE=EXTEND if the information from the virtual filestore of the remote system is not compatible with the entries in the local FTAM catalog for the file that is to be extended.

If FTAM-1 files are mapped to the real filestore of BS2000, the data is converted to EBCDIC.DF.04 format (see code tables). FTAM-3 files are not converted.

Case h:
Transfer a library member from the FTAM partner

In this case, the receive file is a library member.

BS2000	**FTAM filestore of the remote system)**		
Receive file (member type)	document type	universal class	string significance
S	FTAM-1	25 - GraphicString	variable
S	FTAM-1	26 - VisibleString	variable
S	FTAM-1	27 - GeneralString	not-significant
S	FTAM-1	22 - IA5String	not-significant
S	FTAM-3	————	variable
R	not transferable		
C, L	not transferable		
other	FTAM-1	25 - GraphicString	variable
other	FTAM-1	26 - VisibleString	variable
other	FTAM-1	27 - GeneralString	not-significant
other	FTAM-1	22 - IA5String	not-significant
other	FTAM-3	————	variable

In this case, there is no memory for the FTAM-specific file attributes. The default values are assigned, provided this is compatible with the FTAM attributes of the send files.

If FTAM-1 files are mapped to the real filestore of BS2000, the data is converted to EBCDIC.DF.04 format (see code tables). FTAM-3 files are not converted.

3.8.3 Addressing via Application Entity Title (AET)

In the OSI world, communication partners are represented by application entities. An application entity is an addressable entity in Layer 7 of the OSI Reference Model (Application Layer). Such an application entity is the access point of an FTAM application, for example, via which an OSI-TP communication partner can connect to the FTAM application. In the OSI-TP standard, every application entity is assigned to an application entity title, via which the application entity can be addressed uniquely in the OSI network.

Two forms of AET are defined in the ISO Standard, the Directory Form and the Object Identifier Form. openFT-FTAM for BS2000 by default sends a "Nil-Application Entity Title". The FTAM functions of openFT for Unix systems and openFT for Windows support the Object Identifier Form of the AET. An AET comprises two parts:

– Application Process Title (APT)
– Application Entity Qualifier (AEQ).

By default, openFT-FTAM as of V8.0 sends a "Nil-Application Entity Title" as the calling or called Application Entity Title (AET) when performing transfers with the FTAM protocol. The transfer of Nil AETs can be set system-wide via an optional Rep.

The Nil AET is: 1.3.9999.1.7

Addressing FTAM partners with AET

If a called AET is to differ from the "Nil Application Entity Title" then it must be specified in the partner list on instance identification (command: ADD-FT-PARTNER).

The specification has the following syntax:

n1.n2[.n3] [.n10][..m]

n1.n2[.n3] [.n10]
: specifies the *application process title*, between two and ten decimal numbers separated by a period (.). The range and the meaning of the numbers are explained below.

[..m]
: specifies the *application entity qualifier*, range of *m* see below. The two periods are mandatory if a AEQ is specified.

Example

A FTAM partner on computer *daisy2* with APT=*1.0.56.881.4* and AEQ=*785* is to be entered in the partner list under the name *daisyftm*. To do this, enter the following command:

```
ADD-FT-PARTNER PARTNER-NAME=DAISYFTM, -
               PARTNER-ADDRESS=FTAM://DAISY2, -
               IDENTIFICATION=1.0.56.881.4..785
```

Application Process Title (APT)

The APT used to identify the application. The APT should be unique worldwide in accordance with the OSI Standard. For this reason, it should be issued and registered by a Standardization Committee).

An APT in Object Identifier Form is consists of up to 10 components:
`(component1,component2,...,component10)`

The values for component1 to component10 are partially standardized. In this context, a symbolic name was assigned to several numbers. The range of values for component2 depends on the value of component1. The following table shows the symbolic names and the value ranges of the functions supported by FTAM:

component1	component2	component3 to component10
0: CCITT	0: RECOMMENDATION 1: QUESTION 2: ADMINISTRATION 3: NETWORK-OPERATOR (permissible values: 0 - 39)	Permissible values: 0 - 67 108 863
1: ISO	0: STANDARD 1: REGISTRATION-AUTHORITY 2: MEMBER-BODY 3: IDENTIFIED-ORGANIZATION (permissible values: 0 - 39)	Permissible values: 0 - 67 108 863
2:JOINT-ISO-CCITT	Permissible values: 0 - 67 108 863	Permissible values: 0 - 67 108 863

The APT which you specify need not be stipulated by a standardization committee, i.e. you may stipulate your own APT. It must satisfy the following two conditions:

– it must be unique throughout the network
– it must be made up of values that are permissible according to the table above

A remote partner that requests AETs must know this APT in order to set up a connection.

Application Entity Qualifier (AEQ)

The AEQ identifies an access point within an application. You can assign AEQs to the access points of an application only if you have assigned an APT to that application. It is assigned by the operator of the application.

The AEQ is a positive whole number between 0 and 67108863.

You must not use the same AEQ more than once within an application, i.e there must never be two access points with the same AEQ in one application. However, you do not have to assign all the access points in an application to an AEQ.

4 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.

4.1 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 BS2000/OSD - Installation and Administration.

4.1.1 FT command overview

Setting and showing openFT instances

Set openFT instance	SET-FT-INSTANCE	page 255
Show openFT instance	SHOW-FT-INSTANCE	page 282

Transfer files

Submit asynchronous FT request	TRANSFER-FILE	page 361
Submit synchronous FT request	TRANSFER-FILE-SYNCHRONOUS	page 414
Cancel FT requests	CANCEL-FILE-TRANSFER	page 156
Show information on FT requests	SHOW-FILE-TRANSFER	page 260
Modify FT request queue	MODIFY-FILE-TRANSFER	page 207

File management

Show attributes of file/files in a remote system	SHOW-REMOTE-FILE-ATTRIBUTES	page 347
Change file attributes in a remote system	MODIFY-REMOTE-FILE-ATTRIBUTES	page 247
Delete file in a remote system	DELETE-REMOTE-FILE	page 194
Create directory in a remote system	CREATE-REMOTE-DIR	page 185
Change attributes of a directory in a remote system	MODIFY-REMOTE-DIR-ATTR	page 247
Delete directory in a remote system	DELETE-REMOTE-DIR	page 191
Show FTAM attributes of a local file	SHOW-FILE-FT-ATTRIBUTES	page 256
Modify FTAM attributes of a local file	MODIFY-FILE-FT-ATTRIBUTES	page 202

Execute remote commands

Execute commands in the remote system	EXECUTE-REMOTE-CMD	page 197

Logging Function

Show log records or log files	SHOW-FT-LOGGING-RECORDS	page 284

Monitoring

Show monitoring data	SHOW-FT-MONITOR-VALUES	page 303

Obtain information on openFT

Display operating parameters	SHOW-FT-OPTIONS	page 317
Display partner systems	SHOW-FT-PARTNERS	page 328

4.1.2 FTAC commands overview

openFT-AC must be installed in order to use the following commands:

Edit FTAC admission profiles

Create admission profile	CREATE-FT-PROFILE	page 163
Delete admission profile	DELETE-FT-PROFILE	page 188
Modify admission profile	MODIFY-FT-PROFILE	page 219
Show admission profile	SHOW-FT-PROFILE	page 337

Edit FTAC admission sets

Modify admission set	MODIFY-FT-ADMISSION-SET	page 213
Show admission set	SHOW-FT-ADMISSION-SET	page 277

Show partner systems

Display partner systems and security levels	SHOW-FT-RANGE	page 344

4.2 Entering FT commands

Please remember the following when entering commands:

– You must insert commas to separate the individual operands of a command, e.g.

```
/TRANSFER-FILE TRANSFER-DIRECTION=TO,PARTNER=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.

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.

4.3 Command syntax representation

The following example shows the representation of the syntax of a command in a manual. 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.

HELP-SDF Alias: **HPSDF**

GUIDANCE-**MODE** = ***NO** / ***Y**ES
,**SDF-COM**MANDS = ***NO** / ***Y**ES
,**ABBR**EVIATION-**RULES** = ***NO** / ***Y**ES
,**GUID**ED-**DIA**LOG = ***Y**ES(...)
***Y**ES(...)
SCREEN-STEPS = ***NO** / ***Y**ES
,**SPEC**IAL-**FUNC**TIONS = ***NO** / ***Y**ES
,**FUNC**TION-**KEYS** = ***NO** / ***Y**ES
,**NEXT-FIELD** = ***NO** / ***Y**ES
,**UNGUID**ED-**DIA**LOG = ***Y**ES(...) / ***NO**
***Y**ES(...)
SPECIAL-**FUNC**TIONS = ***NO** / ***Y**ES
,**FUNC**TION-**KEYS** = ***NO** / ***Y**ES

Representation of the syntax of the user command HELP-SDF

This syntax description is valid for SDF V4.6A.The syntax of the SDF command/statement language 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. They can be used to limit the set of values (suffix begins with *without*), extend it (suffix begins with *with*), or declare a particular task mandatory (suffix begins with *mandatory*). The following short forms are used in this manual for data type suffixes:

cat-id	cat
completion	compl
correction-state	corr
generation	gen
lower-case	low
manual-release	man
odd-possible	odd
path-completion	path-compl
separators	sep
temporary-file	temp-file
under-score	under
user-id	user
version	vers
wildcard-constr	wild-constr
wildcards	wild

The description of the ‘integer’ data type in Table 3 contains a number of items in italics which are not part of the syntax. They are only used to make the table easier to read. For special data types that are checked by the implementation, Table 3 contains suffixes printed in italics (see the *special* suffix) which are not part of the syntax.

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 $, #, @	
cat-id	A…Z 0…9	Not more than 4 characters; must not begin with the string PUB
command-rest	freely selectable	
composed-name	A…Z 0…9 $, #, @ hyphen period catalog ID	Alphanumeric string that can be split into multiple substrings by means of a period or hyphen. If a file name can also be specified, the string may begin with a catalog ID in the form :cat: (see data type filename).
c-string	EBCDIC character	Must be enclosed within single quotes; the letter C may be prefixed; any single quotes occurring within the string must be entered twice.
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
device	A…Z 0…9 hyphen	Character string, max. 8 characters in length, corresponding to a device available in the system. In guided dialog, SDF displays the valid operand values. For notes on possible devices, see the relevant operand description.

Table 2: Data types (part 1 of 7)

Data type	Character set	Special rules
fixed	+, - 0…9 period	Input format: [sign][digits].[digits] [sign]: + or - [digits]: 0...9 must contain at least one digit, but may contain up to 10 characters (0...9, period) apart from the sign.
filename	A…Z 0…9 $, #, @ hyphen period	Input format: [:cat:][$user.] { file | file(no) | group | group { (*abs) | (+rel) | (-rel) } } :cat: optional entry of the catalog identifier; character set restricted to A...Z and 0...9; maximum of 4 characters; must be enclosed in colons; default value is the catalog identifier assigned to the user ID, as specified in the user catalog. $user. optional entry of the user ID; character set is A…Z, 0…9, $, #, @; maximum of 8 characters; first character cannot be a digit; $ and period are mandatory; default value is the user's own ID. $. (special case) system default ID

Table 2: Data types (part 2 of 7)

Data type	Character set	Special rules
		file file or job variable name; may be split into a number of partial names using a period as a delimiter: $name_1$[.$name_2$[...]] $name_i$ does not contain a period and must not begin or end with a hyphen; file can have a maximum length of 41 characters; it must not begin with a $ and must include at least one character from the range A...Z.

Table 2: Data types (part 3 of 7)

Data type	Character set	Special rules
filename (continued)		#file (special case) @file (special case) # or @ used as the first character indicates temporary files or job variables, depending on system generation. file(no) tape file name no: version number; character set is A...Z, 0...9, $, #, @. Parentheses must be specified. group name of a file generation group (character set: as for “file”) group{(*abs) (+rel) (-rel)} (*abs) absolute generation number (1-9999); * and parentheses must be specified. (+rel) (-rel) relative generation number (0-99); sign and parentheses must be specified.
integer	0…9, +, -	+ or -, if specified, must be the first character.
name	A…Z 0…9 $, #, @	Must not begin with 0...9.

Table 2: Data types (part 4 of 7)

Data type	Character set	Special rules
partial-filename	A…Z 0…9 $, #, @ hyphen period	Input format: [:cat:][$user.][partname.] :cat: see filename $user. see filename partname optional entry of the initial part of a name common to a number of files or file generation groups in the form: $name_1$.[$name_2$.[...]] $name_i$ (see filename). The final character of “partname” must be a period. At least one of the parts :cat:, $user. or partname must be specified.
posix-filename	A...Z 0...9 special characters	String which may have a maximum length of 255 characters. Consists of either one or two periods or of alphanumeric characters and special characters.The special characters must be escaped with a preceding \ (backslash). The / is not allowed. Must be enclosed within single quotes if alternative data types are permitted, separators are used, or the first character is a ?, ! or ^. A distinction is made between uppercase and lowercase.
posix-pathname	A...Z 0...9 special characters structure identifier: slash	Input format: [/]$part_1$/.../$part_n$ where $part_i$ is a posix-filename; maximum of 510 in *POSIX syntax; must be enclosed within single quotes if alternative data types are permitted, separators are used, or the first character is a ?, ! or ^

Table 2: Data types (part 5 of 7)

Data type	Character set	Special rules
product-version	A…Z 0…9 period single quote	Input format: [[C]'][V][m]m.naso['] correction status release status where m, n, s and o are all digits and a is a letter. Whether the release and/or correction status may/must be specified depends on the suffixes to the data type (see the suffixes without-corr, without-man, mandatory-man and mandatory-corr in table 3). product-version may be enclosed within single quotes (possibly with a preceding C). The specification of the version may begin with the letter V.
structured-name	A…Z 0…9 $, #, @ hyphen	Alphanumeric string which may comprise a number of substrings separated by a hyphen. First character: A...Z or $, #, @
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
vsn	a) A…Z 0…9 b) A…Z 0…9 $, #, @	a) Input format: pvsid.sequence-no max. 6 characters pvsid: 2-4 characters; PUB must not be entered sequence-no: 1-3 characters b) Max. 6 characters; PUB may be prefixed, but must not be followed by $, #, @.

Table 2: Data types (part 6 of 7)

Data type	Character set	Special rules
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.
x-text	Hexadecimal: 00…FF	Must not be enclosed in single quotes; the letter X must not be prefixed. There may be an odd number of characters.

Table 2: Data types (part 7 of 7)

Suffixes for data types

Suffix	Meaning
x..y *unit*	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. *unit* with “integer” only: additional units. The following units may be specified: *days* *byte* *hours* *2Kbyte* *minutes* *4Kbyte* *seconds* *Mbyte* milliseconds
x..y *special*	With the other data types: length specification For data types catid, date, device, product-version, time and vsn 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. *special* Specification of a suffix for describing a special data type that is checked by the implementation. “*special*” can be preceded by other suffixes. The following specifications are used: *arithm-expr* arithmetic expression (SDF-P) *bool-expr* logical expression (SDF-P) *string-expr* string expression (SDF-P) *expr* freely selectable expression (SDF-P) *cond-expr* conditional expression (JV) *symbol* CSECT or entry name (BLS)
with	Extends the specification options for a data type.
-compl	When specifying the data type “date”, SDF expands two-digit year specifications in the form yy-mm-dd to: 20yy-mm-dd if yy < 60 19yy-mm-dd if yy ≥ 60
-low	Uppercase and lowercase letters are differentiated.
-path-compl	For specifications for the data type “filename”, SDF adds the catalog and/or user ID if these have not been specified.

Table 3: Data type suffixes (part 1 of 7)

<table>
<tr><th>Suffix</th><th>Meaning</th></tr>
<tr><td>with (cont.)</td><td></td></tr>
<tr><td>-under</td><td>Permits underscores (_) for the data types “name” and “composed-name”.</td></tr>
<tr><td>-wild(n)</td><td>Parts of names may be replaced by the following wildcards.
n denotes the maximum input length when using wildcards.
Due to the introduction of the data types posix-filename and posix-pathname, SDF now accepts wildcards from the Unix world (referred to below as POSIX wildcards) in addition to the usual BS2000 wildcards. However, as not all commands support POSIX wildcards, their use for data types other than posix-filename and posix-pathname can lead to semantic errors.
Only POSIX wildcards or only BS2000 wildcards should be used within a search pattern. Only POSIX wildcards are allowed for the data types posix-filename and posix-pathname. If a pattern can be matched more than once in a string, the first match is used.
<table>
<tr><th>BS2000 wildcards</th><th>Meaning</th></tr>
<tr><td>*</td><td>Replaces an arbitrary (even empty) character string. If the string concerned starts with *, then the * must be entered twice in succession if it is followed by other characters and if the character string entered does not contain at least one other wildcard.</td></tr>
<tr><td>Terminating period</td><td>Partially-qualified entry of a name.
Corresponds implicitly to the string “./*”, i.e. at least one other character follows the period.</td></tr>
<tr><td>/</td><td>Replaces any single character.</td></tr>
<tr><td><s_x:s_y></td><td>Replaces a string that meets the following conditions:
– It is at least as long as the shortest string (s_x or s_y)
– It is not longer than the longest string (s_x or s_y)
– It lies between s_x and s_y in the alphabetic collating sequence; numbers are sorted after letters (A...Z0...9)
– s_x can also be an empty string (which is in the first position in the alphabetic collating sequence)
– s_y can also be an empty string, which in this position stands for the string with the highest possible code (contains only the characters X'FF')</td></tr>
</table>
</td></tr>
</table>

Table 3: Data type suffixes (part 2 of 7)

Suffix	Meaning	
with (cont.)	<s_1,…>	Replaces all strings that match any of the character combinations specified by s. s may also be an empty string. Any such string may also be a range specification "s_x:s_y" (see above).
with-wild(n)		
	-s	Replaces all strings that do not match the specified string s. The minus sign may only appear at the beginning of string s. Within the data types filename or partial-filename the negated string -s can be used exactly once, i.e. -s can replace one of the three name components: cat, user or file.
	Wildcards are not permitted in generation and version specifications for file names. Only system administration may use wildcards in user IDs. Wildcards cannot be used to replace the delimiters in name components cat (colon) and user ($ and period).	
	POSIX wildcards	Meaning
	*	Replaces any single string (including an empty string). An * appearing at the first position must be duplicated if it is followed by other characters and if the entered string does not include at least one further wildcard.
	?	Replaces any single character. It is not permitted as the first character outside single quotes.
	[c_x-c_y]	Replaces any single character from the range defined by c_x and c_y, including the limits of the range. c_x and c_y must be normal characters.
	[s]	Replaces exactly one character from string s. The expressions [c_x-c_y] and [s] can be combined into [s_1c_x-c_ys_2].
	[!c_x-c_y]	Replaces exactly one character not in the range defined by c_x and c_y, including the limits of the range. c_x and c_y must be normal characters. The expressions [!c_x-c_y] and [!s] can be combined into [!s_1c_x-c_ys_2].
	[!s]	Replaces exactly one character not contained in string s. The expressions [!s] and [!c_x-c_y] can be combined into [!s_1c_x-c_ys_2].

Table 3: Data type suffixes (part 3 of 7)

<table>
<tr><th>Suffix</th><th>Meaning</th></tr>
<tr><td>with (cont.)

-wild-constr(n)</td><td>Specification of a constructor (string) that defines how new names are to be constructed from a previously specified selector, i.e., a selection string with wildcards (see also with-wild). n denotes the maximum input length when using wildcards.
The constructor may consist of constant strings and patterns. A pattern (character) is replaced by the string that was selected by the corresponding pattern in the selector.
The following wildcards may be used in constructors:
<table>
<tr><td>Wildcard</td><td>Meaning</td></tr>
<tr><td>*</td><td>Corresponds to the string selected by the wildcard * in the selector.</td></tr>
<tr><td>Termina-ting period</td><td>Corresponds to the partially-qualified specification of a name in the selector.
Corresponds to the string selected by the terminating period in the selector.</td></tr>
<tr><td>/ or ?</td><td>Corresponds to the character selected by the / or ? wildcard in the selector.</td></tr>
<tr><td><n></td><td>Corresponds to the string selected by the n-th wildcard in the selector, where n is an integer.</td></tr>
</table>
Allocation of wildcards to corresponding wildcards in the selector:
All wildcards in the selector are numbered from left to right in ascending order (global index).
Identical wildcards in the selector are additionally numbered from left to right in ascending order (wildcard-specific index).
Wildcards can be specified in the constructor by one of two mutually exclusive methods:
1. Wildcards can be specified via the global index: <n>
2. The same wildcard may be specified as in the selector; substitution occurs on the basis of the wildcard-specific index. For example:
the second “/” corresponds to the string selected by the second “/” in the selector</td></tr>
</table>

Table 3: Data type suffixes (part 4 of 7)

Suffix	Meaning
with-wild-constr(n) (continued)	The following rules must be observed when specifying a constructor: – The constructor can only contain wildcards of the selector. – If the string selected by the wildcard <...> or [...] is to be used in the constructor, the index notation must be selected. – The index notation must be selected if the string identified by a wildcard in the selector is to be used more than once in the constructor. For example: if the selector “A/” is specified, the constructor “A<n><n>” must be specified instead of “A//”. – The wildcard * can also be an empty string. Note that if multiple asterisks appear in sequence (even with further wildcards), only the last asterisk can be a non-empty string, e.g. for “****” or “*//*”. – Valid names must be produced by the constructor. This must be taken into account when specifying both the constructor and the selector. – Depending on the constructor, identical names may be constructed from different names selected by the selector. For example: “A/*” selects the names “A1” and “A2”; the constructor “B*” generates the same new name “B” in both cases. To prevent this from occurring, all wildcards of the selector should be used at least once in the constructor. – If the selector ends with a period, the constructor must also end with a period (and vice versa). The string selected by the terminating period in the constructor cannot be specified via the global index.

Table 3: Data type suffixes (part 5 of 7)

<table>
<tr><th>Suffix</th><th>Meaning</th></tr>
<tr><td>with-wild-constr(n) (continued)</td><td>Examples:
<table>
<tr><th>Selector</th><th>Selection</th><th>Constructor</th><th>New name</th></tr>
<tr><td>A//*</td><td>AB1
AB2
A.B.C</td><td>D<3><2></td><td>D1
D2
D.CB</td></tr>
<tr><td>C.<A:C>/<D,F></td><td>C.AAD
C.ABD
C.BAF
C.BBF</td><td>G.<1>.<3>.XY<2></td><td>G.A.D.XYA
G.A.D.XYB
G.B.F.XYA
G.B.F.XYB</td></tr>
<tr><td>C.<A:C>/<D,F></td><td>C.AAD
C.ABD
C.BAF
C.BBF</td><td>G.<1>.<2>.XY<2></td><td>G.A.A.XYA
G.A.B.XYB
G.B.A.XYA
G.B.B.XYB</td></tr>
<tr><td>A//B</td><td>ACDB
ACEB
AC.B
A.CB</td><td>G/XY/</td><td>GCXYD
GCXYE
GCXY. [1]
G.XYC</td></tr>
</table>
[1] The period at the end of the name may violate naming conventions (e.g. for fully-qualified file names).</td></tr>
<tr><td>without</td><td>Restricts the specification options for a data type.</td></tr>
<tr><td>-cat</td><td>Specification of a catalog ID is not permitted.</td></tr>
<tr><td>-corr</td><td>Input format: [[C]'][V][m]m.na[']
Specifications for the data type product-version must not include the correction status.</td></tr>
<tr><td>-gen</td><td>Specification of a file generation or file generation group is not permitted.</td></tr>
<tr><td>-man</td><td>Input format: [[C]'][V][m]m.n[']
Specifications for the data type product-version must not include either release or correction status.</td></tr>
<tr><td>-odd</td><td>The data type x-text permits only an even number of characters.</td></tr>
<tr><td>-sep</td><td>With the data type “text”, specification of the following separators is not permitted: ; = ( ) < > ␣ (i.e. semicolon, equals sign, left and right parentheses, greater than, less than, and blank).</td></tr>
<tr><td>-temp-file</td><td>Specification of a temporary file is not permitted (see #file or @file under filename).</td></tr>
</table>

Table 3: Data type suffixes (part 6 of 7)

Suffix	Meaning
without (cont.)	
-user	Specification of a user ID is not permitted.
-vers	Specification of the version (see “file(no)”) is not permitted for tape files.
-wild	The file types posix-filename and posix-pathname must not contain a pattern (character).
mandatory	Certain specifications are necessary for a data type.
-corr	Input format: [[C]'][V][m]m.naso['] Specifications for the data type product-version must include the correction status and therefore also the release status.
-man	Input format: [[C]'][V][m]m.na[so]['] Specifications for the data type product-version must include the release status. Specification of the correction status is optional if this is not prohibited by the use of the suffix without-corr.
-quotes	Specifications for the data types posix-filename and posix-pathname must be enclosed in single quotes.

Table 3: Data type suffixes (part 7 of 7)

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).

4.4 Command return codes

The openFT commands supply return codes that you can query when using SDF-P. Each return code consists of a subcode1 (SC1), a subcode2 (SC2) and the maincode (MC).

Subcode1

Subcode1 represents the error class. It is a decimal number. The possible error classes are:

– No error:
  the value of subcode1 is 0.
– Syntax error:
  the value of subcode1 is between 1 and 31, inclusive.
– Internal error (system error):
  the value of subcode1 is 32.
– Errors not assigned to any other class:
  the value of subcode1 is between 64 and 127, inclusive. If the value of subcode 1 is in this range, the maincode must be evaluated in order to ascertain the appropriate action.
– Command cannot be executed at this time:
  the value of subcode1 is between 128 and 130, inclusive.

Subcode2

Subcode2 either contains information supplementary to that in subcode1 or is equal to 0.

Maincode

The maincode corresponds to the message key of the SYSOUT message. You can use the /HELP-MSG-INFORMATION command to fetch detailed information.

For the command return codes of the file transfer and file management commands refer to section “Command return codes for file transfer and file management” on page 433.The command return codes of the remaining commands are always located after the detailed description of the command. In each case, the corresponding section specifies which command return codes are possible and what their meaning is.

4.5 OPS variables

With OPS (Output Presentation Service), you have the option to create the outputs of SHOW commands alternative or additional to the output in SYSLST/SYSOUT in OPS variables. For this to be possible, SDF-P must be installed. The user must generate the corresponding OPS variables with DECLARE-VARIABLE. The information supplied by SHOW commands is stored by openFT in an SDF-P structure, which can be evaluated with the help of an SDF-P procedure. Structure elements which have not been set due to a corresponding command input are output without value assignment.

The request to set OPS variables is made by integrating the unchanged FT command into the BS2000 command EXEC-CMD.

Example

```
/DECLARE-VARIABLE VARIABLE-NAME=<variable-name>,TYPE=*STRUCTURE(...)...
/EXEC-CMD (SHOW-FILE-TRANSFER),TEXT=*N,STRUCT-OUT=<variable-name>
```

The following openFT user commands offer OPS support:

– SHOW-FILE-TRANSFER
– SHOW-FILE-FT-ATTRIBUTES
– SHOW-FT-ADMISSION-SET
– SHOW-FT-INSTANCE
– SHOW-FT-LOGGING-RECORDS
– SHOW-FT-MONITOR-VALUES
– SHOW-FT-OPTIONS
– SHOW-FT-PARTNERS
– SHOW-FT-PROFILE
– SHOW-FT-RANGE
– SHOW-REMOTE-FILE-ATTRIBUTES

4.6 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:

– SHOW-FILE-TRANSFER

– SHOW-FILE-FT-ATTRIBUTES

– SHOW-REMOTE-FILE-ATTRIBUTES

– SHOW-FT-ADMISSION-SET

– SHOW-FT-LOGGING-RECORDS

– SHOW-FT-MONITOR-VALUES

– SHOW-FT-OPTIONS

– SHOW-FT-PARTNERS

– SHOW-FT-PROFILE

– SHOW-FT-RANGE

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.

One example of a possible evaluation procedure is supplied a template in the Microsoft Excel format under the name $SYSFJAM.FTACCNT.XLT. You will need to first make a binary copy of this template on your PC. The template evaluates a CSV log file by means of an automatically running macro. The result shows the number of inbound and outbound requests and the Kilobytes transferred in each case for all BS2000 users.

4.7 CANCEL-FILE-TRANSFER Cancel file transfer requests

Note on usage

User group: FT user and FT administrator

Alias names: CNFT / NCANCEL / FTCANREQ

Functional description

The CANCEL-FILE-TRANSFER 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.

When a request is canceled, it is only deleted completely from the request file after it has been deleted from the request file in the remote system.

Operands that are not shown in the following overview, but which appear in the SDF command syntax using the SHOW command, are only available to the administrator.

Format

CANCEL-**FILE**-**T**RANSFER / **CNFT** / **NCAN**CEL / **FTCANREQ**

```
 TRANSFER-ID = *ALL/ <integer 1..2147483647>
,SELECT = *OWN / *PARAMETERS(...)
    *PARAMETERS(...)
        OWNER-IDENTIFICATION = *OWN / <name 1..8>
        ,INITIATOR = (*LOCAL, *REMOTE) / list-poss(2): *LOCAL / *REMOTE
        ,PARTNER = *ALL / <text 1..200 with-low>
        ,FILE-NAME = *ALL / <filename 1..54> / <c-string 1..512 with-low> /
                          *LIBRARY-ELEMENT(...) /  *POSIX(NAME = <posix-pathname 1..510>))
            *LIBRARY-ELEMENT(...)
                LIBRARY = *ALL / <filename 1..54>
                ,ELEMENT = *ALL / <filename 1..64 without-gen-vers>(...) /
                                  <composed-name 1..64 with-under>(...)
                    <filename>(...) / <composed-name>(...)
                        VERSION = *ALL / <text 1..24>
                ,TYPE = *ALL / <name 1..8>
        ,MONJV = *NONE / <filename 1..54 without-gen-vers>
        ,JV-PASSWORD = *NONE / <c-string 1..4> / <x-string 1..8> /
                               <integer -2147483648..2147483647> / *SECRET
```

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. You can specify either the name of the partner system from the partner list or the address of the partner system, see section "Defining the partner computer" on page 84.

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..54> / <c-string 1..512 with-low> / *POSIX(NAME = <posix-pathname 1..510>)
Cancels FT requests in the local system that access this file.

FILE-NAME = *LIBRARY-ELEMENT(...)
Cancels FT requests that access library members in the local system.

LIBRARY =
Selects the library concerned.

LIBRARY = *ALL
The library name is not used as a selection criterion to determine the FT requests to be canceled.

LIBRARY = <filename 1..54>
FT requests that access this library are to be canceled.

ELEMENT =
Selects the library concerned.

ELEMENT = *ALL
The name of the library member is not a selection criterion to determine the FT requests to be canceled.

ELEMENT = <filename 1..64 without-gen-vers>(...) / <composed-name 1..64 with-under>(...)
Name of the library member concerned.

VERSION =
Version of the library member.

VERSION = *ALL
The version of the library member is not a selection criterion for the FT requests to be canceled.

VERSION = <text 1..24>
Only FT requests that access this version of the library member are to be canceled.

TYPE =
Type of the library member concerned.

TYPE = *ALL
The type of library member is not used as a selection criterion to determine the FT requests to be canceled.

TYPE = <name 1..8>
Only FT requests that access library members of this type are to be canceled.

MONJV =
If appropriate, selects the specific FT request that is being monitored by this job variable.

MONJV = *NONE
A job variable is not selected as a selection criterion to cancel the file transfer.

MONJV = <filename 1..54 without-gen-vers>
The FT monitored by this job variable is to be canceled.

JV-PASSWORD =
If required, specifies the password needed to access the job variable.
If you have already notified the system of the password with the BS2000 command ADD-PASSWORD, you do not have to specify JV-PASSWORD.

JV-PASSWORD = *NONE
The job variable is not password-protected.

JV-PASSWORD = <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
This password is required to access the job variable.

JV-PASSWORD = *SECRET
The system issues the request to enter the password. However, input is not displayed on the screen.

The specification of more than one selection criteria in the CANCEL-FILE-TRANSFER command may result in a file transfer request being “overdefined” (e.g. by entries for TRANSFER-ID and MONJV). If all selection criteria for a request apply, the job is canceled. If not all selection criteria for a request apply, it is not canceled.
If the specified criteria conflict, the CANCEL-FILE-TRANSFER command is acknowledged with the following message:

```
%  FTR0504 No requests available for the selection criteria
```

In such a case there is no jump to the next SET-JOB-STEP in procedures as no error has occurred.

Command return codes

(SC2)	SC1	Maincode	Meaning
0	0	CMD0001	There are no requests that meet the specified selection criteria.
32	32	CMD0221	Request rejected. Internal error. Job variable not accessible.
33	32	CMD0221	Request rejected. Internal error.
36	32	CMD0221	Request rejected. Request data inconsistent.
82	32	CMD0221	Internal error. Job variable not accessible.
83	32	CMD0221	Internal error.
36	64	FTR1036	User not authorized for other user IDs.
38	64	FTR1038	Request is in the termination phase and can no longer be cancelled.
47	64	FTR1047	Request with the specified transfer ID could not be found.
226	64	FTR2226	Job variable contents inconsistent.
227	64	FTR2227	Job variable not in use by openFT.
228	64	FTR2228	Job variable not found.

SC1/2 = Subcode 1/2 in decimal notation
For additional information, see section “Command return codes” on page 153

Example 1

An openFT user wants to cancel all FT requests with his/her user ID. It is enough simply to issue the

```
/CANCEL-FILE-TRANSFER
```

command without specifying any further operands. If only one request is present, openFT acknowledges the request with the following message:

```
%MESS %  FTR2072 Request 229583776 has been canceled
```

If there is more than one request, the following prompt appears first:

```
%  FTR0560 Cancel all specified requests? Reply (y=yes: n=no)
```

After the FT user entered a Y, the following messages appear:

```
%MESS %  FTR2072 Request 23958376 has been canceled
%MESS %  FTR2072 Request 23958461 has been canceled
%MESS %  FTR2072 Request 23958507 has been canceled
```

Example 2

The FT request with the transfer ID 194578 is to be deleted. If the CANCEL-FILE-TRANSFER command is to be issued under the same ID as that under which the FT request was also submitted, the following command is sufficient:

```
/CANCEL-FILE-TRANSFER TRANSFER-ID=194578
```

The recommended short form of this command is as follows:

```
/CNFT 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:

```
/CANCEL-FILE-TRANSFER                                          -
/       SELECT=(INITIATOR=*REMOTE,                             -
/       PARTNER=VAR001,                                        -
/       FILE-NAME=DATA)
```

The recommended short form of this command is as follows:

```
/CNFT SEL=(INIT=*REM,PART-NAME=VAR001,FILE=DATA)
```

or

```
/NCAN SEL=(INIT=*REM,PART-NAME=VAR001,FILE=DATA).
```

4.8 CREATE-FT-PROFILE 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 CREATE-FT-PROFILE to set up their own admission profiles under their user IDs. Users must activate admission profiles predefined by the FTAC administrator with MODIFY-FT-PROFILE (see page 219ff) before they can be used. Profiles predefined by the FTAC administrator may be used immediately if the FTAC administrator also possesses the TSOS 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 BS2000 commands can be specified. For detailed information refer to the section “Preprocessing and postprocessing” on page 96.

Operands which are not explained in the following overview but which appear with a “SHOW” on the SDF command syntax can only be accessed by the administrator.

Format

(part 1 of 2)

CREATE-**FT-PROF**ILE

NAME = ***STD** / <alphanum-name 1..8>

,**PASS**WORD = **_*NONE_** / <c-string 1..8 with-low> / <x-string 1..16> / ***SEC**RET

,**TRANS**FER-**AD**MISSION = ***NOT-SPEC**IFIED / <alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...) / ***SEC**RET

<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_** / ***FIRST** / ***NOT-SPEC**IFIED / ***NONE** / <alphanum-name 1..8>

,**PASS**WORD = **_*OWN_** / <c-string 1..8> / <c-string 9..32> / <x-string 1..16> / ***NONE** / ***SEC**RET

,**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..54 > / <c-string 1..512 with-low> /
***EXP**ANSION(...) / ***LIB**RARY-ELEMENT(...) / ***POS**IX(NAME=<posix-pathname 1..510>

,***EXP**ANSION(...)
PREFIX = <filename 1..53> / <partial-filename 2..53> / <c-string 1..511 with-low>

***LIB**RARY-ELEMENT(...)
LIBRARY = ***NOT-REST**RICTED / <filename 1..54> / ***EXP**ANSION(...)
***EXP**ANSION(...)
PREFIX = <filename 1..53> / <partial-filename 2..53>
,**ELEM**ENT = ***NOT-REST**RICTED / <composed-name 1..64 with-under>(...) / ***EXP**ANSION(...)
<composed-name 1..64 with-under>(...)
VERSION = ***STD** / <text 1..24>
***EXP**ANSION(...)
PREFIX = <composed-name 1..63 with-under> / <partial-filename 2..63>
,**TYPE** = ***NOT-REST**RICTED / <name 1..8>

,**FILE-PASS**WORD = ***NOT-REST**RICTED / ***NONE** / <c-string 1..4> / <x-string 1..8> /
<integer -2147483648...2147483647> / ***SEC**RET

,**PROC**ESSING**-AD**MISSION = ***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..8>
,**PASSW**ORD = ***SAME** / ***NOT-REST**RICTED / ***NONE** / <c-string 1..8> / <c-string 9..32> /
<x-string 1..16> / ***SEC**RET

,**SUCC**ESS-PROCESSING = ***NOT-REST**RICTED / ***NONE** / <c-string 1..1000 with-low> / ***EXP**ANSION(...)

***EXP**ANSION(...)
PREFIX = ***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(...)
PREFIX = ***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 FT profile already exists
```

The command SHOW-FT-PROFILE (see page 337ff) can be used to view the already existing names. To obtain this information, the command SHOW-FT-PROFILE 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 = <c-string 1..8 with-low> / <x-string 1..16>
This FTAC password is required.

PASSWORD = *SECRET
The system prompts you to input the password. However, the password does not appear on the screen.

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 system, 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 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.

TRANSFER-ADMISSION = *SECRET
The system prompts you to input the transfer admission. However, this does not appear on the screen. The operands VALID, USAGE and EXPIRATION-DATE can also be secretly entered in this case.

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 279) 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 279). 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.

USER-ADMISSION = *OWN
For USER-IDENTIFICATION and ACCOUNT, the specifications are taken from the current LOGON authorization. A possible BS2000 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 BS2000.

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 = *FIRST
The first account number assigned to the home pubset of the specified USER-IDENTIFICATION at the time the profile is used in the system is used for account assignment in the case of transfer requests. If the ID's account number changes, the profile does not have to be modified.

ACCOUNT = *NOT-SPECIFIED
No account number is defined.

whose account number he/she does not know**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 admission profile is used.

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.

PASSWORD =
BS2000 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.

PASSWORD = *NONE
No password is required for the user ID specified in the USER-IDENTIFICATION.

PASSWORD = <c-string 1..8> / <c-string 9..32> / <x-string 1..16>
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.

PASSWORD = *SECRET
The system prompts you to enter the password. The entry does not appear on the screen.

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 84. 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..54> / <c-string 1..512 with-low> /
***POSIX(NAME = <posix-pathname 1..510>)**
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 56). When follow-up processing is specified, this file can be referenced with %FILENAME.
You can also directly specify file transfer with file pre- or post-processing here by entering a pipe symbol ’|’ followed by a command.

FILE-NAME = *EXPANSION(PREFIX = <filename 1..53> / <partial-filename 2..53> / <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.

Please note that the part of a DVS 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 96.

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 (‘;’).

There must not be a space between the semicolon and the slash.

Example

```
FILE-NAME = 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) or "..." (double quotes).

Special cases

– A file name or file name prefix that begins with the string 'lftexecsv' must be specified for admission profiles that are to be exclusively used for the ftexec command (see “Example 3” on page 183).

– 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 183).

FILE-NAME = *LIBRARY-ELEMENT(...)
Determines which of your libraries and library members may be accessed by FT requests which use this admission profile.

LIBRARY =
Defines which libraries may be accessed with this admission profile.

LIBRARY = *NOT-RESTRICTED
The admission profile does not restrict access to libraries.

LIBRARY = <filename 1..54>
Only this library may be accessed.

LIBRARY = *EXPANSION(PREFIX = <filename 1..53> / <partial-filename 2..53>)
Only those libraries may be accessed which begin with the specified prefix. FTAC sets the prefix in front of a library name in an FT request which works with this admission profile, and then permits access to the library *Prefix-Libraryname*.

ELEMENT =
Determines which library members may be accessed with this admission profile.

ELEMENT = *NOT-RESTRICTED
Permits unrestricted access to library members.

ELEMENT = <composed-name 1..64 with-under>(...)
Permits access to the specified library member.

VERSION =
Access is only permitted for a specific version of the library member.

VERSION = *STD
Permits access only to the highest version of the library member.

VERSION = <text 1..24>
Access is only permitted for this version of the library member.

ELEMENT = *EXPANSION(PREFIX = <partial-filename 2..63> / <composed-name 1..63 with-under)
Defines a prefix. When a name for a library member is specified in an FT request which works with this admission profile, FTAC adds the specified prefix to this member name. The admission profile then permits access to this member with the name *PrefixMembername*.

TYPE =
Specifies a certain type of library member. The admission profile then only permits access to library members of this type.

TYPE = *NOT-RESTRICTED
Access is not restricted to a certain type of library member.

TYPE = <name 1..8>
FT requests which work with this admission profile may only access library members of this type.

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 = <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
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!

FILE-PASSWORD = *SECRET
The system prompts you to enter the password. However, the password does not appear on the screen.

PROCESSING-ADMISSION =
You can enter a user ID in your BS2000 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.

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. If you wish to perform follow-up processing with FTAM partners, PROCESSING-ADMISSION must have a value other than *NOT-RESTRICTED.

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..8>
Follow-up processing is to be settled under this account number.

PASSWORD =
You specify, where applicable, the BS2000 passwordfor 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.
The entry *SAME is only possible here if the follow-up processing is not started with the /ENTER command.

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 = <c-string 1..8> / <c-string 9..32> / <x-string 1..16>
FT requests which use this admission profile may only initiate follow-up processing on user IDs which are protected with this password.

PASSWORD = *SECRET
The system prompts you to enter the password. The entry does not appear on the screen.

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.

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.

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.

Example

If PREFIX='/PRINT-FILE ' is defined and SUCC='filename' specified in the FT request, then FT executes the command "/PRINT-FILE filename" as 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.
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 = *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 or FTAM 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 44.

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 SHOW-FT-PROFILE.

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('/PRINT-FILE 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 BS2000 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 MONTHLYREPORT.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:

```
/CREATE-FT-PROFILE NAME=GOLDMORE,                                  -
/                    TRANSFER-ADMISSION='monthlyreportfortheboss',  -
/                    IGNORE-MAX-LEVELS=(INBOUND-RECEIVE=*YES,       -
/                    INBOUND-PROCESSING=*YES),                      -
/                    TRANSFER-DIRECTION=*FROM-PARTNER,              -
/                    PARTNER=GOLDMINE,                              -
/                    FILE-NAME=MONTHLYREPORT.GOLDMINE,              -
/                    SUCCESS-PROCESSING=                            -
/                    '/PRINT-FILE␣MONTHLYREPORT.GOLDMINE',          -
/                    FAILURE-PROCESSING=*NONE,                      -
/                    WRITE-MODE=*REPLACE-FILE
```

The short form of this command is:

```
/CRE-FT-PROF␣GOLDMORE,TRANS-AD='monthlyreportfortheboss',          -
/IGN-MAX-LEV=(I-R=*YES,I-P=*YES),TRANS-DIR=*FROM,         -
/PART=GOLDMINE, FILE-NAME=MONTHLYREPORT.GOLDMINE,                  -
/SUCC='/PRINT-FILE␣MONTHLYREPORT.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 BS2000 computer in the file NOTHINGBUTLIES, can use the following openFT command to send it to the central computer JACKJOHN and print it out there:

```
/TRANSFER-FILE␣TO,JACKJOHN,(NOTHINGBUTLIES),
               (FILE=*NOT-SPECIFIED,TRANS-AD='monthlyreportfortheboss')
```

Example 2

A profile is to be created that only allows monitoring.

```
CREATE-FT-PROFILE 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" FTBS2 ONLYFTMONITOR
```

Alternatively, the monitoring values can be output as rows to a file (in this case ftbs2_data), for instance with the following command:

```
ncopy FTBS2!"-po=10" ftbs2_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.

```
/CR-FT-PRO NUR1VORV,,'GetLoggingRecords'            -
,FILE-NAME=*EXP('|ftexecsv -p="/SH-FT-LOG-REC ,"')  -
,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 FTBS2 3 GetLoggingRecords
```

– BS2000 system:

```
/EXE-REM-CMD FTBS2,'3','GetLoggingRecords'
```

– z/OS system:

```
FTEXEC FTBS2,'3','GetLoggingRecords'
```

Command return codes

(SC2)	SC1	Maincode	Meaning
0	0	FTC0051	A user ID with the same name already exists.
0	0	FTC0056	Transfer admission is blocked.
0	64	FTC0100	An FT profile with the same name already exists.
0	64	FTC0101	An FT profile with the specified transfer admission. already exists.
0	64	FTC0150	The access password is missing.
0	64	FTC0153	The owner identification entered is not the own user ID.
0	64	FTC0157	No authorization to create the profile.
0	64	FTC0172	The User-Admission entered does not exist in the system.
0	64	FTC0173	The Processing-Admission entered does not exist in the system.
0	64	FTC0178	The partner name entered occurs several times.
0	64	FTC0182	Maximum length for partner names has been exceeded.
0	64	FTC0200	The total length of the two follow-up processing commands is too long.
0	64	FTC0255	A system error has occurred.

SC1/2 = Subcode 1/2 in decimal notation

For additional information, see section “Command return codes” on page 153

4.9 CREATE-REMOTE-DIR Create remote directory

Note on usage

User group: FT user

Alias name: FTCREDIR

Functional description

With the CREATE-REMOTE-DIR command, you can create a directory in an FT partner system.

Format

CREATE-**REM**OTE-**DIR** / **FTCREDIR**

PARTNER = <text 1..200 with-low>

,**DIR**ECTORY-NAME = **_*NOT-SPECIFIED_** / <filename 1..54> / <c-string 1..512 with-low>

,**PASS**WORD = **_*NONE_** / <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128> /
***SECRET**

,**TRANS**FER-**ADM**ISSION = **_*NONE_** / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> /
***SECRET** / ***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> /
***SECRET**

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 84.

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..54> / <c-string 1..512 with-low>
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 FTR2154.

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.

PASSWORD = *SECRET
The system prompts you to enter the password. However, your input is not displayed on the screen.

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 = *SECRET
The system prompts you to input the transfer admission. However, this is not visible on the screen.

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.

PASSWORD = *SECRET
The system prompts you to enter the password. However, your input is not displayed on the screen.

Command return codes

For a list of the possible return codes, see the table as of page 433.

4.10 DELETE-FT-PROFILE 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 DELETE-FT-PROFILE , 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 337ff), you can view the profiles and decide which ones you no longer need.

Format

DELETE**-FT-PROF**ILE

NAME = ***ALL** / <alphanum-name 1..8> / *STD

,**PASS**WORD = ***NONE** / <c-string 1..8 with-low> / <x-string 1..16> / ***SEC**RET

,**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> / ***SEC**RET

,**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 = <c-string 1..8 with-low> / <x-string 1..16>
Specifies the corresponding FTAC password.

PASSWORD = *SECRET
The system prompts you to enter the password. However, the password does not appear on the screen.

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.

TRANSFER-ADMISSION = *SECRET
The system prompts you to enter the transfer admission. This does not appear on the screen.

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.

Command return codes

(SC2)	SC1	Maincode	Meaning
0	64	FTC0053	No FT profile exists with these criteria.
0	64	FTC0150	The access password is missing.
0	64	FTC0153	The owner identification entered is not the user's own ID.
0	64	FTC0255	A system error occurred.

SC1/2 = Subcode 1/2 in decimal notation
For additional information, see section "Command return codes" on page 153

4.11 DELETE-REMOTE-DIR Delete remote directory

Note on usage

User group: FT user

Alias name: FTDELDIR

Functional description

With the DELETE-REMOTE-DIR command, you can delete a directory in an FT partner system.

Format

DELETE-**REM**OTE-**DIR** / **FTDELDIR**

PARTNER = <text 1..200 with-low>

,**DIR**ECTORY-NAME = ***NOT-SPECIFIED** / <filename 1..54> / <c-string 1..512 with-low>

,**PASS**WORD = ***NONE** / <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128> / ***SECRET**

,**TRANS**FER-**ADM**ISSION = ***NONE** / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> / ***SECRET** / ***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> / ***SECRET**

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 84.

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..54> / <c-string 1..512 with-low>
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 FTR2155.

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.

PASSWORD = *SECRET
The system prompts you to enter the password. However, your input is not displayed on the screen.

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 = *SECRET
The system prompts you to input the transfer admission. However, this is not visible on the screen.

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.

PASSWORD = *SECRET
The system prompts you to enter the password. Your input is not displayed on the screen.

Command return codes

For a list of the possible return codes, see the table as of page 433.

Example:

Delete the empty directory `Dir1` on the Unix system `partux` under the transfer admission `transadm`.

```
DELETE-REMOTE-DIRftdeldir partux,c'Dir1',,transadm
```

4.12 DELETE-REMOTE-FILE Delete remote files

Note on usage

User group: FT user

Alias name: FTDEL

Functional description

The DELETE-REMOTE-FILE command can be used to delete a file in an FT partner system.

Format

DELETE-**REM**OTE-**FILE** / **FTDEL**

PARTNER = <text 1..200 with-low>

,**FILE** = ***NOT-SPECIFIED** / <filename 1..54> / <c-string 1..512 with-low>

,**PASS**WORD = ***NONE** / <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128> /
***SECRET**

,**TRANS**FER-**ADM**ISSION = ***NONE** / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> /
***SECRET** / ***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> /
***SECRET**

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 84

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..54> / <c-string 1..512 with-low>
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.
If the file name is specified with unattached Public Volume Set (BS2000/OSD), the request is rejected with the message FTR2155.

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.

PASSWORD = *SECRET
The system requests you to enter the password. However, the input is not displayed on the screen.

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 = *SECRET
The system requests you to enter the transfer admission. However, the input is not displayed on the screen.

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.

PASSWORD = *SECRET
The system requests you to enter the password. The input is not displayed on the screen.

Command return codes

For a list of the possible return codes, see the table as of page 433.

Example

From your BS2000 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.

```
/DELETE-REMOTE-FILE PARTNER=HUGO,FILE-NAME=FILE, -
/                   TRANSFER-ADMISSION=DELETE-ACCESS
```

Short form:

```
/DEL-REM-FI HUGO,FILE,,DELETE-ACCESS
```

4.13 EXECUTE-REMOTE-CMD Execute remote command

Note on usage

User group: FT user

Alias name: FTEXEC

Functional description

With the EXECUTE-REMOTE-CMD 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 *SYSLST, *SYSOUT or to a file.

EXECUTE-REMOTE-CMD is only available for openFT partners and FTAM partners from Fujitsu Technology Solutions.

The exit code, i.e. the result of the command is output in the local system as subcode 2 of the EXECUTE-REMOTE-CMD command. If the received exit code exceeds the value range of the local exit code (BS2000 systems only have a 1-byte exit code whereas Windows systems have 4-byte exit codes), then the content of the lower-order byte is output.

If the command is not executed in the remote system then a transfer command exit code is output at STDOUT and EXECUTE-REMOTE-CMD terminates with exit code 255. The meaning of the exit code is system-specific.

In the case of output to *SYSLST, it is possible to specify character sets.

In the case of output to *SYSOUT, the character set specified in the local BS2000/OSD is used.

Format

EXECUTE-**REM**OTE-**CMD** / **FTEXEC**

PARTNER = <text 1..200 with-low>

,**CMD**= ***NOT-SPECIFIED** / <c-string 1..400 with-low> (...)
  CODED-CHARACTER-**SET** = ***STD** / <name 1..8>

,**TRANS**FER-**ADM**ISSION = ***NONE** / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> /
***SECRET** / ***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> /
***SECRET**

,**OUT**PUT = ***SYSOUT** / ***SYSLST** / ***FILE**(...)

***FILE**(...)
  FILE-NAME = <filename 1..54>
  ,**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 84.

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 = <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 = *SECRET
The system prompts you to input the transfer admission. However, this is not visible on the screen.

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.

PASSWORD = *SECRET
The system prompts you to enter the password. However, your input is not displayed on the screen.

OUTPUT =
Specifies where the data generated by the command should be output following transfer in the local system.
If the partner is a BS2000 system, output to SYSLST from the remote command is redirected to the channel specified here. Output to SYSOUT is always shown locally on SYSOUT.

OUTPUT = *SYSOUT
The data is written to *SYSOUT.

OUTPUT = *SYSLST
The data is written to *SYSLST.

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..54>
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 XHCS 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.

Command return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
108	128	FTR0108	Request rejected. Remote system not accessible.
4	1	CMD0202	The selected parameters could not be specified simultaneously.
33	32	CMD0221	Request rejected. Internal error.
36	32	CMD0221	Request rejected. Inconsistent request data.
83	32	CMD0221	Internal error.
51	64	FTR2051	Encryption not possible for this request.
125	128	FTR2125	Request rejected. Transport connection lost.
169	64	FTR2169	Request rejected. Remote system: Transfer admission invalid. Transfer admission incorrect or missing FTAC admissions.
170	64	FTR2170	Request rejected. Remote system: Function not supported.
rc	64	FTR2207	The command returned an error in the remote system. The exit code of the remote command can be queried using subcode 2 (rc).

SC1/2 = subcode 1/2 in decimal format
For additional information refer to the section “Command return codes” on page 153.

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
```

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.

4.14 MODIFY-FILE-FT-ATTRIBUTES Modify local FT file attributes

Note on usage

User group: FT user

Alias name: FTMODF

The command is only useful in conjunction with FTAM functionality.

Functional description

The MODIFY-FILE-FT-ATTRIBUTES command is used to modify the FTAM attributes of a file in a local system so that they are suitable for a file transfer request or file management request with an FTAM partner. Invalid combinations of attributes are rejected with the message FTR2018. Values can be assigned to the following attributes:

– file access rights for an FTAM partner that the FTAM partner cannot change (PERMITTED-ACTIONS)
– file type (DATA-TYPE)
– character set (CHARACTER-SET)
– record format (RECORD-FORMAT)
– record length (RECORD-SIZE)

File attributes file type, character set and record format may only be changed if you are aware of the file contents. If this is not the case, file inconsistencies may occur, with the result that data transfer requests to the affected file are terminated, see page 104.

Please note that when you use MODIFY-FILE-FT-ATTRIBUTES, you do not negate the BS2000 file attributes. This means that you can still delete the file with BS2000 resources (e.g. ERASE-FILE), even if the PERMITTED-ACTIONS do not permit deletion for an FTAM partner.

Format

MODIFY−**FILE-FT-ATTR**IBUTES / **FTMODF**

FILE-NAME = <filename 1..54>

,**PASS**WORD = ***NONE** / <integer -2147483648..2147483647> / <c-string 1..4> / <x-string 1..8> / ***SECRET**

,**PERM**ITTED-**ACT**IONS = ***UNCHA**NGED / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

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

,**TRANS**FER-**ATTR**IBUTES = ***UNCHA**NGED / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

DATA-TYPE = ***UNCHA**NGED / ***BIN**ARY / ***CHAR**ACTER(...)

***CHAR**ACTER(...)

CHARACTER-**SET** = ***GRAPHIC** / ***GENERAL** / ***IA5** / ***VISIBLE**

,**REC**ORD-**FORM**AT = ***UNCHA**NGED

,**REC**ORD-**SIZE** = ***UNCHA**NGED / <integer 1..32767>

Operands

FILE-NAME = <filename 1..54>
File in the local system whose attributes are to be modified. These attributes only apply for partners who wish to transfer files with the use of FTAM functionality.
With regard to file transfer admission, the same rules apply as for the BS2000 MODIFY-FILE-ATTRIBUTES command.
If the file name is specified with unattached Public Volume Set, the request is rejected with the error message FTR0020.

PASSWORD =
Password authorizing the user to access the file in the local system. If the file in the local system is password-protected, the password must be specified in this operand unless you notified BS2000 of the password beforehand with ADD-PASSWORD. The password is required in BS2000 for modifying file attributes.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <integer -2147483648..2147483647> / <c-string 1..4> / <x-string 1..8>
Password authorizing the user to access the file in the local system.

PASSWORD = *SECRET
The system prompts you to enter the password. However, the input is not displayed on the screen.

PERMITTED-ACTIONS =
The permitted file accesses. Defines how an FTAM partner can access a local file exclusively. The operand does not affect the access rights of a BS2000 file. It merely defines the access rights for a partner who wishes to access a file using FTAM protocols. The FTAM partner cannot modify this definition.

PERMITTED-ACTIONS = *UNCHANGED
The access rights remain unchanged.

PERMITTED-ACTIONS = *PARAMETERS(...)
Changes file access permissions.

READ-FILE = *NO / *YES
The file cannot or can be read.

INSERT-DATA-UNIT = *NO / *YES
Data units, such as records, cannot or can be inserted in the file.

REPLACE-FILE = *NO / *YES
The file cannot or can be overwritten.

EXTEND-FILE = *NO / *YES
The file cannot or can be extended.

ERASE-DATA-UNIT = *NO / *YES
Data units, such as records, cannot or can be deleted from the file.

READ-ATTRIBUTES = *NO / *YES
The file attributes cannot or can be read.

CHANGE-ATTRIBUTES = *NO / *YES
The file attributes cannot or can be changed.

DELETE-FILE = *NO / *YES
The file cannot or can be deleted.

TRANSFER-ATTRIBUTES =
Attributes used for file transfer.

TRANSFER-ATTRIBUTES = *UNCHANGED
The previous attributes remain unchanged.

TRANSFER-ATTRIBUTES = *PARAMETERS(...)

DATA-TYPE =
File type. Specifies whether the file is to be transferred as a text file or a binary file.

DATA-TYPE = *UNCHANGED
The previous file type remains unchanged.

DATA-TYPE = *BINARY
The file is to be transferred as a binary file.

DATA-TYPE = *CHARACTER(...)
The file is to be transferred as a text file.

CHARACTER-SET =
Character set for the text file.

CHARACTER-SET = *GRAPHIC
The file can contain characters from the GO set defined in ISO646 or ISO8859-1, or from the G1 defined in ISO8859-1.

CHARACTER-SET = *GENERAL
The file can contain characters from the C0 set defined in ISO646, the G0 set defined in ISO646 or ISO8859-1, or the G1 set defined in ISO8859-1.

CHARACTER-SET = *IA5
The file can contain characters from the C0 set and the G0 set defined in ISO646.

CHARACTER-SET = *VISIBLE
The file can contain characters from the G0 set defined in ISO646.

In the case of text files with CHARACTER-SET=*GRAPHIC or CHARACTER-SET=*VISIBLE, the data is transferred in the form of variable length records (for SAM-V files), or fixed length records (SAM-F files).

In the case of text files with CHARACTER-SET=*GENERAL or CHARACTER-SET=*IA5, every record is terminated with a CRLF (Carriage Return Line Feed) during file transfer. The transfer units do not necessarily correspond to the real records.

RECORD-FORMAT =
The format of the records in the file. This operand is intended for future extensions.

RECORD-FORMAT = *UNCHANGED
The previous record length remains unchanged.

RECORD-SIZE =
The length of the longest record in the file.

RECORD-SIZE = *UNCHANGED
The previous record format remains unchanged.

RECORD-SIZE = <integer 1..32767>
Record length in bytes with which the data is to be transferred to an FTAM partner. RECORD-SIZE defines the maximum length for transfer units for files with CHARACTER-SET=*IA5, CHARACTER-SET=*GENERAL as well as for SAM-U files.

For DATA-TYPE and CHARACTER-SET, you can only select the combinations that correspond to the file contents and the SAM record format of the file:

Entries for	in SAM record format	DATA-TYPE	CHARACTER-SET	RECORD-FORMAT, displayed in SHOW-FILE-FT-ATTR
Text files	F V F V V V	*CHARACTER *CHARACTER *CHARACTER *CHARACTER *CHARACTER *CHARACTER	*GRAPHIC *GRAPHIC *VISIBLE *VISIBLE *GENERAL *IA5	*FIXED *VARIABLE *FIXED *VARIABLE *UNDEFINED *UNDEFINED
Structured binary files	V	*BINARY	No entry	*VARIABLE
Unstructured binary files	U	*BINARY	No entry	*UNDEFINED

File access errors are also possible if the record length defined for a SAM-F file differs from that defined in the BS2000 catalog.

Command return codes

For a list of the possible return codes, see the table as of page 433.

Example

You wish to reset the access rights of the local file MYFILE, such that FTAM partners have only read access.

```
/MODIFY-FILE-FT-ATTRIBUTES␣FILE-NAME=MYFILE, -
/                          PERMITTED-ACTIONS=(READ-FILE=*YES, -
/                          READ-ATTRIBUTES=*YES,CHANGE-ATTRIBUTES=*NO)
```

Short form:

```
/MOD-FI-FT-AT MYFILE,,(Y,,,,,Y,N)
```

4.15 MODIFY-FILE-TRANSFER Modify request queue

Note on usage

User group: FT user and FT administrator

Alias name: FTMODREQ

Functional description

You use the MODIFY-FILE-TRANSFER 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.

MODIFY-FILE-TRANSFER 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.

Operands which can not be found in the following overview but which appear when you call up “SHOW” for the SDF command syntax are only accessible to the administrator.

Format

MODIFY-**FILE-T**RANSFER / **FTMODREQ**

```
 TRANSFER-ID = *ALL / <integer 1..2147483647>
,SELECT = *OWN / *PARAMETERS(...)
    *PARAMETERS(...)
        OWNER-IDENTIFICATION = *OWN / <name 1..8>
        ,PARTNER = *ALL / <text 1..200 with-low>
        ,FILE = *ALL / <filename 1..54> / <c-string 1..512 with-low> / *LIBRARY-ELEMENT(...)
            *LIBRARY-ELEMENT(...)
                LIBRARY = *ALL / <filename 1..54>
                ,ELEMENT = *ALL / <filename 1..64 without-gen-vers>(...) /
                                  <composed-name 1..64 with-under>(...)
                    <filename>(...) / composed-name 1..64>(...)
                        VERSION = *ALL / <text 1..24>
                ,TYPE = *ALL / <name 1..8>
        ,MONJV = *NONE / <filename 1..54>
        ,JV-PASSWORD = *NONE / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647> /
                       *SECRET
,QUEUE-POSITION = *UNCHANGED / *FIRST / *LAST
,PRIORITY = *UNCHANGED / *NORMAL / *HIGH / *LOW
```

Operands

TRANSFER-ID =
Transfer ID of the outbound request to be modified.

TRANSFER-ID = *ALL
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 84.

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 SHOW-FILE-TRANSFER 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..54> / <c-string 1..512 with-low>
Modifies outbound requests which access this file (DVS/POSIX) in the local system.

FILE = *LIBRARY-ELEMENT(...)
Modifies outbound requests which access library members in the local system.

LIBRARY =
Selects the library.

LIBRARY = *ALL
The library name is not selected as a criterion for the outbound requests to be modified

LIBRARY = <filename 1..54>
Outbound requests are to be modified which access this library.

ELEMENT =
Library member.

ELEMENT = *ALL
The name of the library member is not selected as a criterion for the outbound requests to be modified.

ELEMENT = <filename 1..64 without-gen-vers>(...) / <composed-name 1..64 with-under>(...)
Name of the library member.

VERSION =
Version of the member.

VERSION = *ALL
The library member version is not selected as a criterion for the outbound requests to be modified.

VERSION = <text 1..24>
Only outbound requests which access this version of the library member are to be modified.

TYPE =
Type of library member.

TYPE = *ALL
The member type is not selected as a criterion for the outbound requests to be modified.

TYPE = <name 1..8>
Only outbound requests which access library members of this type are to be modified.

MONJV =
Selects any outbound request which is monitored by this job variable.

MONJV = *NONE
No job variable is used as a selection criterion for outbound requests to be changed.

MONJV = <filename 1..54>
The outbound request monitored by this job variable is to be modified.

JV-PASSWORD =
Password which is needed to access the job variable.
If you have already entered the password using the BS2000 command ADD-PASSWORD, you do not need to enter JV-PASSWORD.

JV-PASSWORD = *NONE
The job variable is not password-protected or it does not need to be specified.

JV-PASSWORD = <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
This password is required for the job variable.

JV-PASSWORD = *SECRET
The system prompts you to enter the password. The entry does not appear on the screen. However, the password does not appear on the screen.

QUEUE-POSITION =
New position of the outbound request that is to be modified in the openFT request queue. The position of an FTAM request can only be changed relative to the requests that affect the same FTAM partner.

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.

Command return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
0	0	CMD0001	There are no requests that meet the specified selection criteria.
32	32	CMD0221	Request rejected. Internal error. Job variable not accessible.
33	32	CMD0221	Request rejected. Internal error.
36	32	CMD0221	Request rejected. Request data inconsistent.
82	32	CMD0221	Internal error. Job variable not accessible.
83	32	CMD0221	Internal error.
36	64	FTR1036	User not authorized for other user IDs.
47	64	FTR1047	Request with the specified transfer ID could not be found.
226	64	FTR2226	Job variable contents inconsistent.
227	64	FTR2227	Job variable not in use by openFT.
228	64	FTR2228	Job variable not found.

SC1/2 = Subcode 1/2 in decimal notation
For additional information, see section “Command return codes” on page 153

Example

```
/SHOW-FILE-TRANSFER
% TRANS-ID  INI  STATE  PARTNER  DIR   BYTE-COUNT FILE-NAME
% 54483612  LOC   WAIT  UNIX1    FROM  0          FILE1
% 11164324  LOC   WAIT  UNIX2    FROM  0          FILE2

/MODIFY-FILE-TRANSFER SELECT=(FILE=FILE2),QUEUE-POS=*FIRST

/SHOW-FILE-TRANSFER
% TRANS-ID  INI  STATE  PARTNER  DIR   BYTE-COUNT FILE-NAME
% 11164324  LOC   WAIT  UNIX2    FROM  0          FILE2
% 54483612  LOC   WAIT  UNIX1    FROM  0          FILE1
```

4.16 MODIFY-FT-ADMISSION-SET 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 MODIFY-FT-ADMISSION-SET 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 /SHOW... 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.

Operands which cannot be found in the following overview but which appear when you call up “SHOW” for the SDF command syntax are only accessible to the FTAC administrator.

Format

MODIFY-**FT-ADM**ISSION-SET

USER-IDENTIFICATION = ***OWN** / <alphanum-name 1..8>

,**PASS**WORD = ***NONE** / <c-string 1..8 with-low> / <x-string 1..16> / ***SEC**RET

,**SEL**ECT-PARAMETER = ***ALL**

,**NEW-PASS**WORD = ***OLD** / ***NONE** / <c-string 1..8 with-low> / <x-string 1..16> / ***SEC**RET

,**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 = <alphanum-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 = <c-string 1..8 with-low> / <x-string 1..16>
This password authorizes this user to use FTAC commands.

PASSWORD = *SECRET
The system prompts you to enter the password. However, the password does not appear on the screen.

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 SHOW... 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 = <c-string 1..8 with-low> / <x-string 1..16>
Specification of the new FTAC password.

NEW-PASSWORD = *SECRET
The system prompts you to input the password. The input does not appear on the screen, however.

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 SHOW-FT-ADMISSION-SET.

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.

```
/SHOW-FT-ADMISSION-SET
```

Short form:

```
/SHOW-FT-ADM
```

He receives the following output:

```
%               MAX. USER LEVELS                MAX. ADM LEVELS
% 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.

```
/MODIFY-FT-ADMISSION-SET MAX-LEVELS = *PARAMETERS(INBOUND-PROCESSING = 0)
```

The short form of this command is

```
/MOD-FT-ADM MAX-LEV = (IN-PROC = 0)
```

He outputs his admission set once more to double-check.

```
/SHOW-FT-ADM

%               MAX. USER LEVELS                MAX. ADM LEVELS
% 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.

```
/CREATE-FT-PROF FRIENDS,TRANS-AD = 'for my friends', IGN-MAX-LEV = (IN-
PROC = *YES)
```

Command return codes

(SC2)	SC1	Maincode	Meaning
0	0	FTC0050	The set security level exceeds the administrator's limit and will remain invalid until the administrator's limit is raised accordingly.
0	64	FTC0150	The authorization password is missing.
0	64	FTC0151	Only the administrator or owner is permitted to make this modification.
0	64	FTC0152	The user ID entered is not the user's own user ID.
0	64	FTC0175	The operand "NEW-PASSWORD" may not be entered for *STD.
0	64	FTC0176	The user ID entered does not exist in the system.
0	64	FTC0255	A system error occurred.

SC1/2 = Subcode 1/2 in decimal notation

For additional information, see section "Command return codes" on page 153

4.17 MODIFY-FT-PROFILE 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 MODIFY-FT-PROFILE 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 SHOW-FT-PROFILE INF=*ALL (LAST-MODIF). The timestamp is also updated if you do not change the properties of the profile, i.e. if you enter MODIFY-FT-PROFILE with the parameter NAME without specifying other parameters.

Operands which are not found in the following overview but which appear with a “SHOW” of the SDF command syntax are only accessible to the FTAC administrator.

Format

(part 1 of 3)

MODIFY**-FT-PROF**ILE

NAME = ***ALL** / ***STD** / <alphanum-name 1..8>

,**PASS**WORD = ***NONE** / <c-string 1..8 with-low> / <x-string 1..16> / ***SEC**RET

,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> / ***SEC**RET
,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>(...) / ***SEC**RET

***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**

(part 2 of 3)

```
,USER-ADMISSION = *UNCHANGED / *OWN / *PARAMETERS(...)

  *PARAMETERS(...)
    | USER-IDENTIFICATION = *OWN / <name 1..8>
    |,ACCOUNT = *OWN / *FIRST / *NOT-SPECIFIED / *NONE / <alphanum-name 1..8>
    |,PASSWORD = *OWN / <c-string 1..8> / <c-string 9..32> / <x-string 1..16> / *NONE / *SECRET

,INITIATOR = *UNCHANGED / list-poss(2): *REMOTE / *LOCAL

,TRANSFER-DIRECTION = *UNCHANGED / *NOT-RESTRICTED / *FROM-PARTNER / *TO-PARTNER

,PARTNER = *UNCHANGED / *NOT-RESTRICTED / *ADD(...) / *REMOVE(...) /
                    list-poss(50): <text 1..200 with-low>

  *ADD(...)
    | NAME = list-poss(50): <text 1..200 with-low>

  *REMOVE(...)
    | NAME = list-poss(50): <text 1..200 with-low>

,MAX-PARTNER-LEVEL = *UNCHANGED / *NOT-RESTRICTED / <integer 0..100>

,FILE-NAME = *UNCHANGED / *NOT-RESTRICTED  / <filename1..54 > /
             <c-string 1..512 with-low> /  *EXPANSION(...) / *LIBRARY-ELEMENT(...) /
             *POSIX(NAME=<posix-pathname 1..510>)

  *EXPANSION(...)
    | PREFIX = <filename 1..53> / <partial-filename 2..53> / <c-string 1..511 with-low>

  *LIBRARY-ELEMENT(...)
    | LIBRARY = *UNCHANGED / *NOT-RESTRICTED / <filename 1..54> / *EXPANSION(...)
    |   *EXPANSION(...)
    |     | PREFIX = <filename 1..53> / <partial-filename 2..53>
    |,ELEMENT = *UNCHANGED / *NOT-RESTRICTED /
    |                 <composed-name 1..64 with-under>(...) / *EXPANSION(...)
    |    <composed-name 1..64 with-under>(...)
    |     | VERSION = *STD / <text 1..24>
    |   *EXPANSION(...)
    |     | PREFIX = <composed-name 1..63 with-under> / <partial-filename 2..63>
    |,TYPE = *UNCHANGED / *NOT-RESTRICTED / <name 1..8>

,FILE-PASSWORD = *UNCHANGED / *NOT-RESTRICTED / *NONE / <c-string 1..4> /
                 <x-string 1..8> / <integer -2147483648...2147483647> / *SECRET
```

(part 3 of 3)

```
,PROCESSING-ADMISSION = *UNCHANGED / *SAME / *NOT-RESTRICTED / *PARAMETERS(...)
    *PARAMETERS(...)
        USER-IDENTIFICATION = *SAME / *NOT-RESTRICTED / <name 1..8>
        ,ACCOUNT = *SAME / *NOT-RESTRICTED / *NONE / <alphanum-name 1..8>
        ,PASSWORD = *SAME / *NOT-RESTRICTED / *NONE / <c-string 1..8> / <c-string 9..32> /
                    <x-string 1..16> / *SECRET
,SUCCESS-PROCESSING = *UNCHANGED / *NOT-RESTRICTED / *NONE / <c-string 1..1000 with-low> /
                      *EXPANSION(...)
    *EXPANSION(...)
        PREFIX = *UNCHANGED / *NOT-RESTRICTED / <c-string 1..999 with-low>
        ,SUFFIX = *UNCHANGED / *NOT-RESTRICTED / <c-string 1..999 with-low>
,FAILURE-PROCESSING = *UNCHANGED / *NOT-RESTRICTED / *NONE / <c-string 1..1000 with-low> /
                      *EXPANSION(...)
    *EXPANSION(...)
        PREFIX = *UNCHANGED / *NOT-RESTRICTED / <c-string 1..999 with-low>
        ,SUFFIX = *UNCHANGED / *NOT-RESTRICTED / <c-string 1..999 with-low>
,WRITE-MODE = *UNCHANGED / *NOT-RESTRICTED / *NEW-FILE / *REPLACE-FILE / *EXTEND-FILE
,FT-FUNCTION = *UNCHANGED / *NOT-RESTRICTED / list-poss(5):
               *TRANSFER-FILE / *MODIFY-FILE-ATTRIBUTES / *READ-DIRECTORY /
               *FILE-PROCESSING
,USER-INFORMATION = *UNCHANGED / *NONE / <c-string 1..100 with-low>
,DATA-ENCRYPTION = *UNCHANGED / *NOT-RESTRICTED / *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 = <c-string 1..8 with-low> / <x-string 1..16>
This FTAC password is required.

PASSWORD = *SECRET
The system prompts you to enter the password. However, it does not appear on the screen.

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.

TRANSFER-ADMISSION = *SECRET
The system prompts you to enter the transfer admission. However, it does not appear on the screen.

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   FT profile already exists
```

The command SHOW-FT-PROFILE (see page 337ff) can be used to obtain information on the already existing name. For this information, it suffices to enter SHOW-FT-PROFILE 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   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.

TRANSFER-ADMISSION = *SECRET
The system prompts you to input the transfer admission. However, this does not appear on the screen. The operands VALID, USAGE and EXPIRATION-DATE can also be secretly entered in this case.

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 279).

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 CREATE-FT-PROFILE 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 BS2000 password is only taken from your LOGON authorization when an FT request accesses the admission profile.

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 BS2000.

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 = *FIRST
The first account number assigned to the home pubset of the specified USER-IDENTIFICATION at the time the profile is used in the system is used for account assignment in the case of transfer requests. If the ID's account number changes, the profile has not to be modified.

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.

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..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 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.

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 = <c-string 1..8> / <c-string 9..32> / <x-string 1..16>
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.

PASSWORD = *SECRET
The system prompts you to enter the password. However, this does not appear on the screen.

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 84. 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..54> / <c-string 1..512 with-low> /
***POSIX(NAME = <posix-pathname 1..510>)**
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 56). When follow-up processing is specified, this file can be referenced with %FILENAME.
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..53> /
<partial-filename 2..53> / <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.

Please note that the part of a DVS 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 96.

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 (‘;’).
There must not be a space between the semicolon and the slash.

Example

```
FILE-NAME = 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) or "..." (double 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 CREATE-FT-PROFILE , “Example 3” on page 183).

– 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 183).

FILE-NAME = *LIBRARY-ELEMENT(...)
Determines which of your libraries and library members may be accessed by FT requests which use this admission profile.

LIBRARY =
Defines which libraries may be accessed with this admission profile.

LIBRARY = *UNCHANGED
The library specifications in the admission profile remain unchanged.

LIBRARY = *NOT-RESTRICTED
The admission profile does not restrict access to libraries.

LIBRARY = <filename 1..54>
Only this library may be accessed.

LIBRARY = *EXPANSION(PREFIX = <composed-name 1..63 with-under> / <partial-filename 2..63>)
Only those libraries may be accessed which begin with the specified prefix. FTAC sets the prefix in front of a library name in an FT request which uses this admission profile, and then permits access to the library *Prefix-Libraryname*.

ELEMENT =
Determines which library members may be accessed with this admission profile.

ELEMENT = *UNCHANGED
The library member specifications in the admission profile remain unchanged.

ELEMENT = *NOT-RESTRICTED
Permits unrestricted access to library members.

ELEMENT = <composed-name 1..64 with-under>(...)
Only permits access to the specified library member.

VERSION =
Access is only permitted for a specific version of the library member.

VERSION = *STD
Permits access only to the highest version of the library member.

VERSION = <text 1..24>
Access is only permitted for this version of the library member.

ELEMENT = *EXPANSION(PREFIX = <composed-name 1..63 with-under> / <partial-filename 2..63>)
Defines a prefix. When a name for a library member is specified in an FT request which uses this admission profile, FTAC adds the specified prefix to this member name. The admission profile then permits access to this member with the name *PrefixElementname*.

TYPE =
Specifies a certain type of library member. The admission profile then only permits access to library members of this type.

TYPE = *UNCHANGED
Any access restrictions to individual member types remain unchanged.

TYPE = *NOT-RESTRICTED
Access is not restricted to a certain type of library member.

TYPE = <name 1..8>
FT requests which use this admission profile may only access library members of this type.

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 = <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
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!

FILE-PASSWORD = *SECRET
The system prompts you to enter the password. However, this does not appear on the screen.

PROCESSING-ADMISSION =
You can enter a user ID in your BS2000 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.

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. The entry *SAME is only possible here if the follow-up processing is not started with the command /ENTER.

PROCESSING-ADMISSION = *NOT-RESTRICTED
FT requests which use this admission profile may contain any PROCESSING-ADMISSION. For follow-up processing with FTAM partners, PROCESSING-ADMISSSION must have a value not equal to *NOT-RESTRICTED.

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..8>
Follow-up processing is to be settled under this account number.

PASSWORD =
Specifies, where applicable, the BS2000 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. The entry *SAME is only possible here if the follow-up processing is not started with the command /ENTER.

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 = <c-string 1..8> / <c-string 9..32> / <x-string 1..16>
FT requests which use the admission profile may only initiate follow-up processing on user IDs which are protected with this password.

PASSWORD = *SECRET
The system prompts you to enter the password. The entry does not appear on the screen.

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. 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.

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='/PRINT-FILE ' is defined and SUCC='filename' specified in the FT request, then FT executes the command “/PRINT-FILE filename” as follow-up processing.

– If SUFFIX='filename' is defined and SUCC='/PRINT-FILE' specified in the FT request, then FT executes the command “/PRINT-FILE filename” as 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>
BS2000 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 or FTAM 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 44.

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 SHOW-FT-PROFILE 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('/PRINT-FILE 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:

```
/MODIFY-FT-PROFILE␣NAME = profil1,
          FILE-NAME = *EXPANSION(PREFIX = branch.)
```

A possible short form of this command is:

```
/MOD-FT-PROF␣profil1,FILE-N = (PRE = branch.)
```

This places heavy restrictions on the admission profile. The other specifications remain unchanged.

Command return codes

(SC2)	SC1	Maincode	Meaning
0	0	FTC0051	A user ID with the same name already exists.
0	64	FTC0053	No FT profile exists which meets the criteria specified.
0	64	FTC0055	The partner restrictions were lifted.
0	0	FTC0056	Transfer admission is blocked.
0	64	FTC0100	An FT profile with this name already exists.
0	64	FTC0101	An FT profile with the specified transfer admission already exists.
0	64	FTC0150	The access password is missing.
0	64	FTC0151	Modifications can only be made by the administrator or owner.
0	64	FTC0153	The owner ID entered is not the user's own ID.
0	64	FTC0170	The partner entered is unknown within the partner system available for this user.
0	64	FTC0171	The profile entered does not exist.
0	64	FTC0172	The user admission entered does not exist in the system.
0	64	FTC0173	The processing admission entered does not exist in the system.
0	64	FTC0174	The parameters "NEW-NAME" and "TRANSFER-ADMISSION" may only used together in conjunction with unique selection criteria ("NAME" or "TRANSFER-ADMISSION").
0	64	FTC0178	The partner name entered occurs several times.
0	64	FTC0179	The maximum number of partner restrictions has been exceeded.
0	64	FTC0182	The maximum length of partner names has been exceeded.
0	64	FTC0200	The total length of the two follow-up processing commands is too long.
0	64	FTC0255	A system error has occurred.

SC1/2 = Subcode 1/2 in decimal notation

For additional information, see section "Command return codes" on page 153

4.18 MODIFY-REMOTE-DIR-ATTR Modify remote directory attributes

Note on usage

User group: FT user

Alias name: FTMODDIR

Functional description

With the MODIFY-REMOTE-DIR-ATTR 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

MODIFY-**REM**OTE-**DIR**-**ATTR** / **FTMODDIR**

PARTNER = <text 1..200 with-low>

,**DIR**ECTORY-NAME = ***NOT-SPECIFIED** / <filename 1..54> / <c-string 1..512 with-low>

,**PASS**WORD = ***NONE** / <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128> /
***SECRET**

,**TRANS**FER-**ADM**ISSION = ***NONE** / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> /
***SECRET** / ***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> /
***SECRET**

,**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 84.

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..54> / <c-string 1..512 with-low>
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 an unmounted Public Volume Set then the request is rejected with error message FTR2155.

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.

PASSWORD = *SECRET
The system prompts you to enter the password. However, your input is not displayed on the screen.

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 = *SECRET
The system prompts you to input the transfer admission. However, this is not visible on the screen.

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.

PASSWORD = *SECRET
The system prompts you to enter the password. However, your input is not displayed on the screen.

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.

Command return codes

For a list of the possible return codes, see the table as of page 433.

4.19 MODIFY-REMOTE-FILE-ATTRIBUTES Modify remote file attributes

Note on usage

User group: FT user

Alias name: FTMOD

Functional description

The MODIFY-REMOTE-FILE-ATTRIBUTES command is used to modify the attributes of a file in an FT partner system.

Depending on the partner involved, the following file attributes can be modified:

openFT partners:

– File name
– Access rights

FTAM partners:

– File name
– Availability of the file
– Account number for file storage costs
– Future file size
– Access rights
– Legal qualifications related to the use of a file

FTP partners:

– File name

Format

MODIFY-**REM**OTE-**FILE**-**ATTR**IBUTES / **FTMOD**

PARTNER = <text 1..200 with-low>

,**FILE** = ***NOT-SPECIFIED** / <filename 1..54> / <c-string 1..512 with-low>

,**PASS**WORD = ***NONE** / <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128> /
***SECRET**

,**TRANS**FER-**ADM**ISSION = ***NONE** / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> /
***SECRET** / ***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> /
***SECRET**

,**NEW-NAME** = ***SAME** / <filename 1..54> / <c-string 1..512 with-low>

,**FILE-AVAIL**ABILITY = ***UNCHA**NGED / ***IMMED**IATE / ***DEFERRED**

,**STOR**AGE-**ACC**OUNT = ***UNCHA**NGED / <c-string 1..40 with-low> / <text 1..40>

,**FUTURE-FILE**-SIZE = ***UNCHA**NGED / <integer 1..2147483647>

,**ACCESS-MODE** = ***UNCHA**NGED / ***READ-ONLY** / ***READ-WRITE** / ***REPL**ACE**-ALL-**BY(...) / ***ADD**(...)

***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

***ADD**(...)
- **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 / <c-string 1..80 with-low> / <text 1..80>

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 84.

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..54> / <c-string 1..512 with-low>
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 FTR2155.

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.

PASSWORD = *SECRET
The system prompts you to enter the password. However, the input is not displayed on the screen.

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 = *SECRET
The system prompts you to enter the transfer admission. However, the input is not displayed on the screen.

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.

PASSWORD = *SECRET
The system prompts you to enter the password. However, the input is not displayed on the screen.

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 =
Provides information on the availability of a file in an FTAM partner system. This operand has two possible values: *file available immediately* or *file not available immediately. File not available immediately*, for example could refer to a file stored in an archive. However, the partner can freely interpret *not available immediately*. That is why the conventions of the FTAM partner must be observed in this case.

FILE-AVAILABILITY = *UNCHANGED
The previous file availability remains unchanged.

FILE-AVAILABILITY = *IMMEDIATE
In a remote system, the file attribute is assigned the value *available immediately*.
In the case of requests with openFT and FTAM partners that do not support the storage group, the request is rejected.

FILE-AVAILABILITY = *DEFERRED
In a remote system, the file attribute is assigned the value *not available immediately*. The file can then be stored in the partner system.
Requests involving openFT or FTAM partners that do not support the storage group or this attribute are rejected.

STORAGE-ACCOUNT =
Account number for file storage.

STORAGE-ACCOUNT = *UNCHANGED
The previous account number remains unchanged.

STORAGE-ACCOUNT = <c-string 1..40 with-low> / <text 1..40>
Identifies as storage account for the FTAM partner. The file storage costs are debited to this account, insofar as the partner system supports this function. This operand must conform to the conventions of the partner system.
Requests involving openFT or FTAM partners that do not support the storage group or this attribute are rejected.

FUTURE-FILE-SIZE =
Required future file size.

FUTURE-FILE-SIZE = *UNCHANGED
The previous file size remains unchanged.

FUTURE-FILE-SIZE = <integer 1..2147483647>
Provides FTAM partners with information on the possible future file size. This information acts as a guideline for a system specific optimization.
Requests involving openFT or FTAM partners that do not support the storage group or this attribute are rejected.

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.
With FTAM partners, the access rights that are to be replaced must not be linked with any further specifications, such as, for example a file password.

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.

ACCESS-MODE = *ADD(...)
In the case of FTAM partners, the file receives an additional set of access rights. This entry is only relevant for FTAM partners that support more than one set of 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 =
Legal qualifications.

LEGAL-QUALIFICATION = *UNCHANGED
The previous legal qualifications remain unchanged.

LEGAL-QUALIFICATION = <c-string 1..80 with-low> / <text 1..80>
In the case of FTAM partners, this defines a new legal qualification for a file (similar to a copyright). This must not exceed 80 characters.
Requests involving openFT and FTAM partners that do not support the security group or this attribute are rejected.

Command return codes

For a list of the possible return codes, see the table as of page 433.

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'

```
/MODIFY-REMOTE-FILE-ATTRIBUTES␣PARTNER = HUGO,FILE-NAME = MYFILE,      -
/                              TRANSFER-ADMISSION = (JIM,A1234FT,C'PWD'),-
/                              ACCESS-MODE = *READ-ONLY
```

Short form:

```
/MOD-REM-FI-ATT HUGO,MYFILE,,(JIM,A1234FT,'PWD'),,,,,*R-O
```

4.20 SET-FT-INSTANCE Set an openFT instance

Note on usage

User group: FT user

Functional description

Using this command, you select the openFT instance you would like to immediately begin working with. After successful execution of the SET-FT-INSTANCE command, all the FT and FTAC commands (with the exception of instance commands) and all calls on the program interface are processed by the specified instance.

If you do not set an instance, then you will always work with the default instance. It is the first instance to be displayed on executing the /SHOW-FT-INSTANCE INST=*ALL command.

Format

SET-FT-INSTANCE
NAME = <alphanum-name 1..8>

Operands

NAME = <alphanum-name 1..8>
The name of the openFT instance to which all subsequent FT-/FTAC commands and program interface calls should be directed.

Command return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
83	32	CMD0221	Internal error.
25	64	FTR1025	Instance not found.
27	64	FTR1027	Config user ID not accessible.

SC1/2 = Subcode 1/2 in decimal notation
For additional information, see section “Command return codes” on page 153.

4.21 SHOW-FILE-FT-ATTRIBUTES Display local FT file attributes

Note on usage

User group: FT User

Alias name: FTSHWF

The command is only useful in conjunction with FTAM functionality.

Functional description

With the SHOW-FILE-FT-ATTRIBUTES command you can display the FTAM attributes of a file in the local system. Thus, you can determine the file attribute values for file transfer and file management requests involving FTAM partners.

There are three options for outputting the attributes:

– Display the file name
– Display default information
– Display all file attributes

Format

SHOW-FILE-FT-ATTRIBUTES / **FTSHWF**

FILE-NAME = <filename 1..54>

,**INF**ORMATION = ***STD** / ***ALL-ATTR**IBUTES / ***NAME-ONLY**

,**OUT**PUT = ***SYSOUT**(...) / ***SYSLST**(...)

***SYSOUT**(...) / ***SYSLST**(...)
LAYOUT = ***STD** / ***CSV**

Operands

FILE-NAME = <filename 1..54>
File in the local system whose attributes are to be displayed. These attributes only apply to FTAM partners wishing to use FTAM functionality to transfer files.
The same access rules apply as for the SHOW-FILE-ATTRIBUTES command in BS2000.
If the file name is specified with an unattached Public Volume Set, the request is rejected with the error message FTR0020.

INFORMATION =
Amount of information required. Only those attributes available for file transfer and file management requests are displayed.

INFORMATION = *STD
The default range of information is to be output.

INFORMATION = *ALL-ATTRIBUTES
All available information is output to the file.

INFORMATION = *NAME-ONLY
Only the file name is output.

OUTPUT = *SYSOUT(...)
Output is to SYSOUT.

OUTPUT = *SYSLST(...)
Output is to SYSLST.

LAYOUT = *STD
Output is put into a user-friendly form for reading.

LAYOUT = *CSV
Output occurs 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 page 155).

Command return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
20	64	FTR0020	The command was not executed. Send file unknown.
33	32	CMD0221	Request rejected. Internal error.
36	32	CMD0221	Request rejected. Request data inconsistent.
83	32	CMD0221	Internal error.
170	64	FTR2170	Request rejected. Remote system: Function not supported.

SC1/2 = Subcode 1/2 in decimal notation
For additional information, see section "Command return codes" on page 153.

OPS variables

The following table shows the OPS variables of the command SHOW-FILE-FT-ATTRIBUTES with the operand INF=*ALL-ATTRIBUTES, the underlined values are valid for the output with the operand INF=*STD. For the operand input INF=*NAMES-ONLY only the element F-NAME (String) is output.

Element	Type	Output
F-NAME	String	
STOR-ACCOUNT	String	
CRE	Struct	
.USER	String	
.DATE	String	yyyy-mm-dd
.TIME	String	hh:mm:ss
MODIFY	Struct	
.USER	String	
.DATE	String	yyyy-mm-dd
.TIME	String	hh:mm:ss
READ	Struct	
.USER	String	
.DATE	String	yyyy-mm-dd
.TIME	String	hh:mm:ss
MOD-ATTR	Struct	
.USER	String	
.DATE	String	yyyy-mm-dd
.TIME	String	hh:mm:ss
DATA-TYPE	String	*CHAR / *BINARY / *DIR / *NO-INFO
CHAR-SET [1]	String	*GRAPHIC / *GENERAL / *IA5 / *VISIBLE
REC-FORM	String	*VAR / *FIXED / *UNDEF
REC-SIZE	Integer	
F-AVAIL	String	*IMMED / *DEFERRED
ACCESS-RIGHTS	Struct	
.READ-F	String	*YES / *NO
.INS-DATA-UNIT	String	*YES / *NO
.REPLACE-F	String	*YES / *NO
.EXTEND-F	String	*YES / *NO

Element	Type	Output
.ERASE-DATA-UNIT	String	*YES / *NO
.READ-ATTR	String	*YES / *NO
.MOD-ATTR	String	*YES / *NO
.DEL-F	String	*YES / *NO
.TRAVERSAL	String	*YES / *NO
.REV-TRAVERSAL	String	*YES / *NO
.RANDOM	String	*YES / *NO
F-SIZE	Integer	
MAX-F-SIZE	Integer	
LEGAL-QUALIFICATION	String	
CCS-NAME	String	Value

[1] The element CHAR-SET is only assigned if DATA-TYPE=*CHARS.

Examples

1. You wish to output detailed information on the FTAM attributes of the LOCFILE on the local BS2000 system.

```
/SHOW-FILE-FT-ATTRIBUTES␣FILE-NAME = LOCFILE

%*r-pxeacd--- MAIER 1234567890 Apr 30 11:55 LOCFILE
```

   Short form:

```
/SH-FI-FT LOCFILE
```

2. You wish to output detailed information on the FTAM attributes of the LOCFILE on the local BS2000 system.

```
/SHOW-FILE-FT-ATTRIBUTES␣FILE-NAME=LOCFILE,INFORMATION=*ALL-ATTRIBUTES

%FILENAME=LOCFILE
%CRE   MAIER DATE=Nov 19 11:55
%MOD   DATE=Apr 28 15:54
%REA   DATE=Apr 30 09:01
%RECORD-FORMAT=v
%ACCESS=RIGHTS=r-pxeacd---   FILESIZE=123456
```

   Short form:

```
/SH-FI-FT-AT LOCFILE,*A
```

4.22 SHOW-FILE-TRANSFER Query status of file transfer request

Note on usage

User group: FT user and FT administrator

Alias names: SHFT / NSTATUS / FTSHWREQ

Functional description

The SHOW-FILE-TRANSFER command allows you to request information about FT requests. As with CANCEL-FILE-TRANSFER , 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 SHOW-FILE-TRANSFER 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 (or library member) to be transferred in the local system.
– the number of bytes transferred

By entering INFORMATION=*ALL in the SHOW-FILE-TRANSFER 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 274.

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

SHOW-FILE-TRANSFER / **SHFT** / **NSTAT**US / **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..54> / <c-string 1..512 with-low> /
***LIB**RARY-ELEMENT(...) / ***POS**IX(**NAME**=<posix-pathname 1..510>)
***LIB**RARY-ELEMENT(...)
LIBRARY = ***ALL** / <filename 1..54>
,ELEMENT = ***ALL** / <filename 1..64 without-gen-vers>(...) /
<composed-name 1..64 with-under>(...)
<filename>(...) / composed-name>(...)
VERSION = ***ALL** / <text 1..24>
,TYPE = ***ALL** / <name 1..8>
,MONJV = ***NONE** / <filename 1..54 without-gen-vers>
,JV-PASSWORD = ***NONE** / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647> /
***SECRET**
,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 = ***SYSOUT**(...) / ***SYSLST**(...)
***SYSOUT**(...) / ***SYSLST**(...)
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 84.

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..54> / <c-string 1..512 with-low> / *POSIX(NAME = <posix-pathname 1..510>)
Name of a file. Information is required on the file transfer requests that access this file.

FILE-NAME = *LIBRARY-ELEMENT(...)
Information is required on file transfer requests that access library members in the local system.

LIBRARY =
Selects the library concerned.

LIBRARY = *ALL
The library name is not used as a selection criterion to define the file transfer requests on which information is to be output.

LIBRARY = <filename 1..54>
Name of a library. Information is required on the file transfer requests that access this library.

ELEMENT =
Library member. Information is required on all the file transfer requests that access this member.

ELEMENT = *ALL
The name of the library member is not used as a selection criterion to define the file transfer requests.

ELEMENT = <filename 1..64 without-gen-vers>(...) / <composed-name 1..64 with-under>(...)
Name of a library member. Information is required on the file transfer requests that access this library member.

VERSION =
Version number of the library member.

VERSION = *ALL
Information is required on all file transfer requests that access any version of the library member.

VERSION = <text 1..24>
Information is required on the file transfer requests that access a specific version of the library member.

TYPE =
The type of library member.

TYPE = *ALL
The member type is not used as a selection criterion to define the file transfer requests on which information is to be output.

TYPE = <name 1..8>
Information is required only on those file transfer requests that access library members of this type.

MONJV =
If appropriate, selects the specific file transfer request that is being monitored by this job variable.

MONJV = *NONE
A job variable is not used as a selection criterion to define the file transfer request on which information is to be output.

MONJV = <filename 1..54 without-gen-vers>
Information is required on the file transfer request that is being monitored by this job variable.

JV-PASSWORD =
If required, specifies the password needed to access the job variable.

If you have already notified the system of the password with the BS2000 command ADD-PASSWORD, you do not have to specify JV-PASSWORD.

JV-PASSWORD = *NONE
The job variable is not password-protected.

JV-PASSWORD = <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
This password is required for the job variable.

JV-PASSWORD = *SECRET
The system requests you to enter the password. This input is not displayed on the screen.

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 openFT and FTAM partners. 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 271):
– 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 TRANSFER-FILE command (see “Description of the long output” on page 272).

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 276).

OUTPUT =
Output medium.

OUTPUT = *SYSOUT(...)
Output is sent to SYSOUT.

OUTPUT = *SYSLST(...)
Output is sent to SYSLST.

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 155).

If selection criteria are specified in the SHOW-FILE-TRANSFER command and no request is found that matches all the specified criteria, the command is acknowledged with the following message:

```
% FTR0504 No requests available for the selection criteria
```

In such a case, procedures do not branch to the next SET-JOB-STEP.

Commando return codes

(SC2)	SC1	Maincode	Meaning
0	0	CMD0001	There are no requests that meet the specified selection criteria.
33	32	CMD0221	Request rejected. Internal error.
36	32	CMD0221	Request rejected. Request data inconsistent.
82	32	CMD0221	Internal error. Job variable not accessible.
83	32	CMD0221	Internal error.
88	32	CMD0221	Error during OPS generation.
36	64	FTR1036	User not authorized for other user IDs
47	64	FTR1047	The request with the specified transfer ID could not be found.
226	64	FTR2226	Job variable contents inconsistent.
227	64	FTR2227	Job variable not in use by openFT.
228	64	FTR2228	Job variable not found.

SC1/2 = Subcode 1/2 in decimal notation
For additional information, see section "Command return codes" on page 153

OPS variables

The following table shows the OPS variables for the command SHOW-FILE-TRANSFER with the operand INF=*ALL. The underlined values are valid for the output with the operand INF=*STD. The table on page 270 shows the OPS variables for the output with the operand INF=*SUMMARY.

Element	Type	Output
TRANS-ID	Integer	
STA	String	*SUSPEND / *LOCK / *WAIT / *ACTIVE / *FINISH / *HOLD
BYTE-COUNT	Integer	
PRIO	String	*NORM / *HIGH / *LOW
INIT	String	*LOC / *REM
TRANS-DIRECT	String	*TO-PARTNER / *FROM-PARTNER
PARTNER-NAME	String	
COMPRESS	String	*NONE / *BYTE-REPETITION / *ZIP
DATA-ENC	String	*YES / *NO
DICHECK	String	*YES / *NO
WRITE-MODE	String	* REPL-FILE / *NEW-FILE / *EXT-FILE
FILE-SIZE	String	Value
REC-SIZE	String	Value
REC-FORMAT	String	*STD / *VARIABLE / *FIXED /*UNDEFINED
START	Struct	
.DATE	String	*SOON / yyyy-mm-dd
.TIME	String	*SOON / hh:mm:ss
CANCEL	Struct	
.DATE	String	*NO / yyyy-mm-dd
.TIME	String	*NO / hh:mm:ss
OWNER	String	
DATA-TYPE	String	*CHAR / *BINARY / *NOT-SPEC
TRANSP	String	*YES / *NO
LOC-PAR	Struct	
.F-TYPE [1]	String	*FILE / *LIB
.F-NAME	String	
.LIB	String	
.ELEM	String	

Element	Type	Output
.VERSION	String	
.TYPE	String	
.TRANS-ADMIS	Struct	
.USER-ID	String	
.ACCOUNT	String	
.PROF-NAME [2]	String	
.PROCESS-ADMIS	Struct	
.USER-ID	String	
.ACCOUNT	String	
.SUCC-PROCESS	String	*SECRET / success-processing
.FAIL-PROCESS	String	*SECRET / failure-processing
.LISTING	String	*NONE / *SYSLST / *LISTFILE / *FAIL-SYSLST / *FAIL-LISTFILE
.MONJV	String	
.CCS-NAME	String	*STD / value
REM-PAR	Struct	
.F-TYPE [1]	String	*FILE / *LIB
.F-NAME	String	
.LIB	String	
.ELEM	String	
.VERSION	String	
.TYPE	String	
.TRANS-ADMIS	Struct	
.USER-ID [3]	String	*REM-PROF / user-id
.ACCOUNT [3]	String	*REM-PROF / account
.PROCESS-ADMIS	Struct	
.USER-ID	String	
.ACCOUNT	String	
.SUCC-PROCESS	String	*SECRET / success-processing
.FAIL-PROCESS	String	*SECRET / failure-processing
.CCS-NAME	String	*STD / value

Element	Type	Output
TARGET	Struct	
.FILE-FORMAT	String	*SAME / *BLOCK / *SEQ
.REC-FORMAT	String	*SAME / *UNDEF
PROTECTION	String	*STD / *SAME
GLOBAL-REQ-ID	Integer	

[1] For F-Type=*FILE, LIB, ELEM, VERSION and TYPE are not displayed.

[2] USER-ID and ACCOUNT are not assigned if an FTAC profile is specified.

[3] Since this cannot be output when a remote FTAC transfer admission is specified, USER-ID and ACCOUNT are assigned with *REM-PROFILE in this case.

The following table shows the OPS variables for the output with the operand INF=*SUMMARY.

Element	Type	Output
NUM-ACTIVE	Integer	
NUM-WAIT	Integer	
NUM-LOCK	Integer	
NUM-SUSPEND	Integer	
NUM-HOLD	Integer	
NUM-FINISHED	Integer	
NUM-SUMM [1]	Integer	

[1] Grand total of all requests including the requests that are still not validated and therefore not counted in any of the other elements.

4.22.1 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:

```
/SHOW-FILE-TRANSFER                                                    -
/                  SELECT=(INITIATOR=*REMOTE,                          -
/                  PARTNER=ALFRED,                                     -
/                  FILE-NAME=DRAISINE,                                 -
/                  STATE=*ACTIVE)
```

The recommended short form of this command is as follows:

```
/SHFT SEL=(INIT=*REM,PART-NAME=ALFRED,FILE=DRAISINE,STATE=*ACT)
```

or

```
/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
```

The information is output to SYSOUT, since this is the default value for the output of inquiry information.

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 and there is no BCMAP). 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

4.22.2 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 SYSLST via the FT request with transfer ID 721212. 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:

```
/SHOW-FILE-TRANSFER                                                    -
/                   TRANSFER-ID=721212,                                -
/                   INFORMATION=*ALL,                                  -
/                   OUTPUT=*SYSLST
```

The recommended short form of this command is as follows:

```
/SHFT 721212,INF=*ALL,OUT=*SYSLST
```

The information output on SYSLST then has the following format, for example:

```
%TRANSFER-ID =721212      STORE  =12-07-11 14:09:35  FILESIZE=40960000
%   STATE    =WAIT        BYTECNT=2117632
%   INITIATOR=LOCAL       TRANS  =TO                 PRIO    =NORM
%   WRITE    =REPLACE     START  =SOON               CANCEL  =NO
%   COMPRESS =BYTE        DATA   =CHAR
%   TRANSP   =NO          ENCRYPT=YES                TABEXP  =NO
%   OWNER    =USER1       DICHECK=NO
%   PARTNER  =WIN01
%   PARTNER-STATE =ACT
%   PARTNER-PRIO  =NORM
%   LOC: FILE     =$USER1.FILE.GR
%        TRANS-ADM=(USER1,88888)
%        ASYN-MSG =ALL
%   REM: FILE     =TEST2
%        TRANS-ADM=REMOTE-PROFILE
```

Example 2 (Inbound request)

Full information is to be output to SYSLST on the FT request with transfer ID 983050. 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:

```
/SHOW-FILE-TRANSFER                                                  -
/                  TRANSFER-ID=983050,                               -
/                  INFORMATION=*ALL,                                 -
/                  OUTPUT=*SYSLST
```

```
%TRANSFER-ID =983050      STORE  =12-07-11 14:09:36  FILESIZE=40960000
%   STATE    =WAIT        BYTECNT=1925120
%   INITIATOR=REMOTE      TRANS  =FROM               PRIO    =
%   WRITE    =REPLACE     START  =SOON               CANCEL  =NO
%   COMPRESS =BYTE        DATA   =CHAR               GLOB-ID =721212
%   TRANSP   =NO          ENCRYPT=YES                TABEXP  =NO
%   OWNER    =USER2       DICHECK=YES                RECFORM =VARIABLE
%   PARTNER  =WIN01
%   PARTNER-STATE =ACT
%   PARTNER-PRIO  =NORM
%   FILE     =TEST2
%   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 openFT and FTAM partners (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)

PROTECT: Specifies whether the protection attributes of the file are transferred

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 and there is no BCMAP). 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.

4.22.3 Description of the summary output

You want to output information about the number of request in each individual processing status.

```
/SHFT 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.

4.23 SHOW-FT-ADMISSION-SET 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 SHOW-FT-ADMISSION-SET command to display admission sets. You can output the following information on either SYSOUT or SYSLST:

– 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

SHOW**-FT-ADM**ISSION**-S**ET

USER-IDENTIFICATION = **<u>*OWN</u>** / ***ALL** / ***STD** / <alphanum-name 1..8>

,OUTPUT = **<u>*SYSOUT</u>**(...) / ***SYSLST**(...)

<u>*SYSOUT</u>(...) / ***SYSLST**(...)

LAYOUT = **<u>*STD</u>** / ***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 = <u>*OWN</u>
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 = <alphanum-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.

OUTPUT =
Output medium for the information requested.

OUTPUT = *SYSOUT(...)
Output is sent to SYSOUT.

OUTPUT = *SYSLST(...)
Output is sent to SYSLST.

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 155).

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

```
/SHOW-FT-ADMISSION-SET␣USER-IDENTIFICATION=*ALL

FTSHWADS␣USER-IDENTIFICATION=*ALL
```

Short form:

```
/SHOW-FT-ADM␣*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.

Command return codes

(SC2)	SC1	Maincode	Meaning
0	64	FTC0052	The information output was interrupted.
0	64	FTC0152	The user ID entered is not the user's own ID.
0	64	FTC0181	The FT profile name entered occurs several times.
0	64	FTC0255	A system error occurred.

SC1/2 = Subcode 1/2 in decimal notation

For additional information, see section "Command return codes" on page 153

OPS variables

Element	Type	Output
USER-ID	String	
USER-LEV	Struct	
.MAX-OBS	Integer	
.MAX-OBS-STD	String	*YES / *NO
.MAX-OBR	Integer	
.MAX-OBR-STD	String	*YES / *NO
.MAX-IBS	Integer	
.MAX-IBS-STD	String	*YES / *NO
.MAX-IBR	Integer	
.MAX-IBR-STD	String	*YES / *NO
.MAX-IBP	Integer	
.MAX-IBP-STD	String	*YES / *NO
.MAX-IBF	Integer	
.MAX-IBF-STD	String	*YES / *NO
ADM-LEV	Struct	
.MAX-OBS	Integer	
.MAX-OBS-STD	String	*YES / *NO
.MAX-OBR	Integer	
.MAX-OBR-STD	String	*YES / *NO
.MAX-IBS	Integer	
.MAX-IBS-STD	String	*YES / *NO
.MAX-IBR	Integer	
.MAX-IBR-STD	String	*YES / *NO
.MAX-IBP	Integer	
.MAX-IBP-STD	String	*YES / *NO
PRIV	String	*YES / *NO
.MAX-IBF	Integer	
.MAX-IBF-STD	String	*YES / *NO
PASSWORD	String	*YES / *NO

4.24 SHOW-FT-INSTANCE Display an openFT instance

Note on usage

User group: FT user

Functional description

With the command SHOW-FT-INSTANCE you can display information regarding openFT instances.

Format

SHOW-FT-INSTANCE
INSTANCES = ***CURRENT** / ***ALL**

Operands

INSTANCES =
Scope of the desired information.

INSTANCES = *CURRENT
Information on the currently set openFT instance.

INSTANCES = *ALL
Information on all openFT instances.

OPS variables

The following table shows the OPS variables of the SHOW-FT-INSTANCE command.

Element	Type	Output
NAME	String	
HOST	String	
CONFIG-USERID	String	
AUTO-START	String	*ON / *OFF

Example

```
/SHOW-FT-INST INST=*ALL
%NAME     HOST     CONFIG-USERID   AUTO-START
%STD      -        :V70A:$SYSFJAM  OFF
%TEST     PC17QD3  :V70A:$HUGO     OFF
```

4.25 SHOW-FT-LOGGING-RECORDS Display log records and offline log files

Note on usage

User group: FT user, FT administrator and FTAC administrator

Alias name: FTSHWLOG

Functional description

With the SHOW-FT-LOGGING-RECORDS 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.

SHOW-FT-LOGGING-RECORDS 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! The output can be interrupted using the K2 key.

There are three types of output: short output and long output and CSV format.

FTAC logging

With FTAC functionality, SHOW-FT-LOGGING-RECORDS 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 296.

Format

(part 1 of 2)

SHOW-FT-LOGGING-**REC**ORDS / **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 = **1970-01-01**(...) / <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..54> / <filename-prefix 2..53> /
<c-string 1..512 with-low> / ***DIR**ECTORY(...) / ***POS**IX(NAME=<posix-pathname 1..510>)

***DIR**ECTORY(...)

NAME = ***ALL** / <partial-filename 1..53> / <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..54> / *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 = *SYSOUT(...) / *SYSLST(...)
  *SYSOUT(...) / *SYSLST(...)
      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 = 1970-01-01(...) / <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 84.

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..54> / <c-string 1..512 with-low> / *POSIX(NAME = <posix-pathname 1..510>)
Fully qualified name of the files for which you wish to view the logging records.

FILE-NAME = <filename-prefix 2..53>
Partially qualified name of the files for which you want to view the logging records.

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 SHOW-REMOTE-FILE-ATTRIBUTES (see page 347).

NAME = *ALL
The directory is not a selection criterion

NAME = <partial-filename 1..53> / <c-string 1..512 with-low>
Name of the directory.
In BS2000, directories are represented by partially qualified file names in DVS.

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 openFT and FTAM partners. 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..54>
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 key K2.

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 294).

INFORMATION = *ALL
The logging records are displayed in a detailed format (see page 296).

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 = *SYSOUT(...)
Output is sent to SYSOUT.OUTPUT = *SYSLST(...)
Output is sent to SYSLST.

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 155).

4.25.1 Description of the short output

Short output form of FT logging records (example)

```
/SHOW-FT-LOGGING-RECORDS NUMBER = 2

%TYP LOGG-ID  TIME     RC    PARTNER  INITIATOR INIT USER-ADM FILENAME
%2012-02-26
%T       5333 14:18:24 0014 <G133H301 FT2V292   1TCL FT2V292  TEST2
%T       5284 14:08:12 0000 >G133H301 FT2V292   1TCL FT2V292  TEST1

%TYP LOGG-ID  TIME     RC    PARTNER  INITIATOR INIT USER-ADM FILENAME
%2012-02-26
%T       5333 14:18:24 0014 <G133H301 FT2V292   1TCL FT2V292  TEST2
%T       5284 14:08:12 0000 >G133H301 FT2V292   1TCL 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, P indicates the FTP logging record written by the FTP server from the product „interNet Services in BS2000/OSD“.	
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	

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. 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 BS2000 command HELP-MSG-INFORMATION (FTCxxxx for FTAC type or FTRxxxx for FT type).	
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[1] – request to delete remote files
INITIATOR	Initiator (user ID) in the case of requests issued locally issued; if initiative is from remote system: *REMOTE	
INIT	TSN from which the request came. If the INITIATOR was *REMOTE, the field is empty.	
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.	

[1] When modifying the access rights of a file from an FTAM partner system, two logging records are written. In this case, no direction is specified before the PARTNER output.

4.25.2 Description of the long output

Long output form outbound (example)

```
%LOGGING-ID  = 38735     RC      = 0000        TIME     = 2012-07-11 13:58:21
%   TRANS    = TO        REC-TYPE= FT          FUNCTION = TRANSFER-FILE
%   PROFILE  =           PCMD    = NONE        STARTTIME= 2012-07-11 13:58:21
%   TRANS-ID = 721206    WRITE   = REPLACE     REQUESTED= 2012-07-11 13:58:21
%   TRANSFER =         0 kB                    CCS-NAME =
%   SEC-OPTS = ENCR+DICHK, RSA-1024 / AES-256
%   INITIATOR= TSOS                            INITSN   = 83VV
%   USER-ADM = TSOS
%   PARTNER  = LINUX01
%   FILENAME = $USER1.FILE.TEST

%LOGGING-ID  = 38734     RC      = 0000        TIME     = 2012-07-11 13:58:21
%   TRANS    = TO        REC-TYPE= FTAC        FUNCTION = TRANSFER-FILE
%   PROFILE  =           PRIV    =
%   INITIATOR= TSOS                            INITSN   = 83VV
%   USER-ADM = TSOS
%   PARTNER  = LINUX01
%   FILENAME = $USER1.FILE.TEST
```

Long output form inbound (example)

```
LOGGING-ID   = 38733     RC      = 0000        TIME     = 2012-07-11 13:49:44
%   TRANS    = FROM      REC-TYPE= FT          FUNCTION = TRANSFER-FILE
%   PROFILE  =           PCMD    = NONE        STARTTIME= 2012-07-11 13:49:44
%   TRANS-ID = 721204    WRITE   = REPLACE     STORETIME= 2012-07-11 13:49:44
%   TRANSFER =         1 kB                    CCS-NAME =
%                                              CHG-DATE = SAME
%   SEC-OPTS = ENCR+DICHK+DENCR+DDICHK, RSA-1024 / AES-256
%   INITIATOR= *REMOTE                         GLOB-ID  = 66277
%   USER-ADM = USER1
%   PARTNER  = LINUX01
%   FILENAME = TEST1

LOGGING-ID   = 38732     RC      = 0000        TIME     = 2012-07-11 13:49:44
%   TRANS    = FROM      REC-TYPE= FTAC        FUNCTION = TRANSFER-FILE
%   PROFILE  = PROF1     PRIV    = NO
%   INITIATOR= *REMOTE                         GLOB-ID  = 66277
%   USER-ADM = USER1
%   PARTNER  = LINUX01
%   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 [1] – 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	
PROTECT	Specifies whether the protection attributes are transferred. Is only output if this option was specified in the transfer request.	
	SAME	The protection attributes of the file were 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.	
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 BS2000 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 BS2000 command HELP-MSG-INFORMATION (FTCxxxx for FTAC type or FTRxxxx for FT type).	
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.

Name	Explanation
INITSN	TSN from which the request came, entered only in the case of outbound requests.
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.

[1] When modifying the access rights of a file from an FTAM partner system, two logging records are written. In this case, no direction is specified before the PARTNER output.

Command return codes

(SC2)	SC1	Maincode	Meaning
0	0	CMD0001	No log records available for the selection criteria.
33	32	CMD0221	Request rejected. Internal error.
36	32	CMD0221	Request rejected. Request data inconsistent.
83	32	CMD0221	Internal error.
88	32	CMD0221	Error during OPS generation.
36	64	FTR1036	User not authorized for other user IDs.
2	0	FTR2225	Information output cancelled.

SC1/2 = Subcode 1/2 in decimal notation

For additional information, see section “Command return codes” on page 153

OPS variables

The following table shows the OPS variables for the command SHOW-FT-LOGGING-RECORDS with the operand INF=*ALL. The underlined values are valid for the output with the operand INF=*STD. The output for INF = *LOGGING-FILES has its own format, see page 302.

Depending on the type of log record, not all elements are output.

Element	Type	Output
LOG-ID	Integer	
REASON-CODE [1]	Integer	
LOG	Struct	
.DATE	String	yyyy-mm-dd
.TIME	String	hh:mm:ss
INIT-USER-ID	String	USER-ID of request initiator / *REM
INIT-TSN [2]	String	TSN of request initiator
PARTNER-NAME	String	

Element	Type	Output
TRANS-DIRECT	String	*TO-PARTNER / *FROM-PARTNER / *NOT-SPECIFIED
REC-TYPE	String	*FT / *FTAC
FUNC	String	*TRANS-FILE / *READ-FILE-ATTR / *DEL-FILE / *CRE-FILE / *MOD-FILE-ATTR / *READ-DIR / *CRE-DIR / *MOD-DIR / *DEL-DIR / *MOVE-FILE / *LOGIN
USER-ADMIS	String	
WRITE-MODE	String	*REPL-FILE / *NEW-FILE / *EXT-FILE
RESULT-PROCESS	String	*NONE / *STARTED / *NOT-STARTED
START	Struct	
.DATE	String	yyyy-mm-dd
.TIME	String	hh:mm:ss
TRANS-ID	Integer	
STORE	Struct	
.DATE	String	yyyy-mm-dd
.TIME	String	hh:mm:ss
BYTE-NUM	String / Integer	*NONE / Value
PRIVIL [3]	String	*NO / *YES
PROF-NAME [3]	String	
F-NAME	String	
SEC	Struct	
.PROT.ENC	String	*NO / *YES
.PROT.INT-CHECK	String	*NO / *YES
.USER-DATA.ENC	String	*NO / *YES
.USER-DATA.INT-CHECK	String	*NO / *YES
.LOC-AUTH	String	*NO / *YES
.REM-AUTH	String	*NO / *YES
.AUTH-LEV	Integer	1 / 2 / empty
RSA-KEY-LEN	Integer	
SYMM-ENC-ALG	String	*DES / *AES
PROTECTION [4]	String	*STD / *SAME
ADMINISTRATOR-ID [5]	String	Value
ADM-CMD [5]	String	Value

Element	Type	Output
ROUTING [5]	String	Value
CHANGE-DATE	String	*STD / *SAME
GLOBAL-REQ-ID	Integer	global request identification / empty

[1] The reason code is always given in decimal form. To determine the meaning of FTAC logging records using the manual, the value must be converted to hexadecimal form.

[2] For INIT-USER-ID=*REM, INIT-TSN is not assigned.

[3] Only for REC-TYPE=*FTAC and specification of a profile.

[4] Only with FT log records, not with FTAC or ADM log records

[5] Only for REC-TYPE = ADM

When you specify the INF=*LOGGING-FILES operand, only the two elements below are output:

Element	Type	Output
TIME-STAMP	String	yyyy-mm-dd hh:mm:ss
FILE-NAME	String	Wert

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=.

```
/SHOW-FT-LOGGING-RECORDS 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.

```
/SHOW-FT-LOG-REC (OWN=Meier,CRE-TIME=*INTERVAL(FROM=2012-01-01(00:00),  -
/                           TO=2012-03-31(00:00))),INF=*ALL
```

4.26 SHOW-FT-MONITOR-VALUES Show monitoring data

Note on usage

User group: FT users and FT administrators

Alias: FTSHWMON

Description of the function

The SHOW-FT-MONITOR-VALUES command allows you to output the monitoring values from openFT operation on the local system. To do this, monitoring must be activated (see MODIFY-FT-OPTIONS in the System Administrator Guide) and openFT must be activated.

Format

```
SHOW-FT-MONITOR-VALUES / 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= *SYSOUT(...) / *SYSLST(...)
    *SYSOUT(...) / *SYSLST(...)
        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 314.

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 309).

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. You terminate the command by canceling output by pressing K2.

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 = *SYSOUT(...)
The data is output to SYSOUT.

OUTPUT = *SYSLST(...)
The data is output to SYSLST.

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 155).

If the monitoring configuration changes (filters), the new start time for monitoring is shown in a separate column in CSV format.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
51	32	CMD0221	Internal error.
88	32	CMD0221	Error on OPS output.
1	0	FTR1039	open FT not active
59	64	FTR1059	Monitoring is not active.
2	0	FTR2225	Information output cancelled.

SC1/2 = subcode 1/2 in decimal format
For additional information refer to the section “Command return codes” on page 153.

OPS variables

The following table shows the OPS variables for the SHOW-FT-MONITOR-VALUES command, which are output with the operand NAME = *ALL. Values shown in bold are also output with the operand NAME = *STD.

Element	Type	Output
CURRENT	Struct	
.DATE	String	yyyy-mm-dd
.TIME	String	hh:mm:ss
MON-START	Struct	
.DATE	String	yyyy-mm-dd
.TIME	String	hh:mm:ss
PARTNER-SEL	Struct	
.OPENFT	String	*YES / *NO
.FTAM	String	*YES / *NO
.FTP	String	*YES / *NO
REQUEST-SEL	Struct	
.ASYNC	String	*YES / *NO
.SYNC	String	*YES / *NO
.LOCAL	String	*YES / *NO
.REMOTE	String	*YES / *NO
THROUGHPUT	Struct	
.NET-BYTES-TOTAL	String	Value
.NET-BYTES-SEND	String	Value
.NET-BYTES-RCV	String	Value
.NET-BYTES-TEXT	String	Value

Element	Type	Output
.NET-BYTES-BIN	String	Value
.DISK-TOTAL	String	Value
.DISK-SEND	String	Value
.DISK-RCV	String	Value
.DISK-TEXT	String	Value
.DISK-BIN	String	Value
.REQ-TOTAL	String	Value
.REQ-F-TRANS	String	Value
.REQ-F-MANAG	String	Value
.REQ-SUCC	String	Value
.REQ-ABORT	String	Value
.REQ-INTR	String	Value
.ADMIS-FAIL	String	Value
.FOLLOWUP	String	Value
.CONN-SUCC	String	Value
.CONN-FAIL	String	Value
.CONN-ABORT	String	Value
DURATION	Struct	
.REQ-TOTAL-OUTB	String	Value
.REQ-TOTAL-INB	String	Value
.REQ-F-TRANS-OUTB	String	Value
.REQ-F-TRANS-INB	String	Value
.REQ-F-MANAG-OUTB	String	Value
.REQ-F-MANAG-INB	String	Value
.REQ-WAIT	String	Value
.DNS-OUTB	String	Value
.DNS-INB	String	Value
.CONN-ESTABL	String	Value
.F-OPEN-OUTB	String	Value
.F-OPEN-INB	String	Value
.F-CLOS-OUTB	String	Value
.F-CLOS-INB	String	Value
.ADMIS-CHECK-OUTB	String	Value

Element	Type	Output
.ADMIS-CHECK-INB	String	Value
STATE	Struct	
.NUM-REQ-ACT-ASYN	String	Value
.NUM-REQ-ACT-SYN	String	Value
.NUM-REQ-WAIT	String	Value
.NUM-REQ-HOLD	String	Value
.NUM-REQ-SUSPEND	String	Value
.NUM-REQ-LOCK	String	Value
.NUM-REQ-FINISH	String	Value
.CONN-LIM	String	Value
.NUM-CONN-ACT	String	Value
.REQ-LIM	String	Value
.NUM-REQ-QUEUE	String	Value
.OPENFT-ACT	String	Value
.FTAM-ACT	String	Value
.FTP-ACT	String	Value
.TRACE	String	Value

4.26.1 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 MODIFY-FT-OPTIONS ...,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.

4.26.2 Examples

1. Monitoring values are to be output in default output format.

```
/SHOW-FT-MONITOR-VALUES
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 309 for the meanings.

2. Only the data types are to be output in default output format.

```
/SHOW-FT-MONITOR-VALUES 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).

```
/SHOW-FT-MONITOR-VALUES 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.

4.27 SHOW-FT-OPTIONS Display operating parameters

Note on usage

User group: FT user and FT administrator

Alias name: FTSHWOPT

Functional description

The command SHOW-FT-OPTIONS 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
- Name of the BCAM host
- 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, SNMP traps, ADM traps)
- Settings for the monitoring functions

Format

SHOW-FT-OPTIONS / **FTSHWOPT**
OUTPUT = ***SYSOUT**(...) / ***SYSLST**(...) ***SYSOUT**(...) / ***SYSLST**(...) **LAYOUT** = ***STD** / ***CSV** / ***BS2-PROC** / ***ZOS-PROC**

Operands

OUTPUT =
Output medium.

OUTPUT = *SYSOUT(...)
Output takes place on SYSOUT.

OUTPUT = *SYSLST(...)
Output takes place on SYSLST.

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 155).

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.
If this output is redirected to a file using the SYSFILE command, you should note that the BS2000 SYSFILE management prefixes each line with the space (printer feed control character). The first column of the file must therefore be removed before the procedure generated in this way can be called.

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.

Command return codes

(SC2)	SC1	Maincode	Meaning
83	32	CMD0221	Internal error.
88	32	CMD0221	Error during OPS generation.
35	64	FTR1035	Command only permissible for FT administrator.
2	0	FTR2225	Information output canceled.

SC1/2 = Subcode 1/2 in decimal notation

For additional information, see section “Command return codes” on page 153

OPS variables

The following table shows the OPS variables for the command SHOW-FT-OPTIONS.

Element	Type	Output
REQ-LIM	Integer	
TASK-LIM	String	
CONN-LIM	Integer	
TRANSPORT-UNIT-SIZE	Integer	
PARTNER-CHECK	String	*STD / *TRANSP-ADDR
SEC-LEV	String	Value / *BY-PARTNER-ATTRIBUTES
TRACE	Struct	
.STATE	String	*ON / *OFF
.OUT	String	*FILE / empty
.PARTNER-SEL	Struct	
.OPENFT	String	*YES / *NO
.FTAM	String	*YES / *NO
.FTP	String	*YES / *NO
.ADM	String	*YES / *NO
.REQUEST-SEL	Struct	
.SYNC	String	*YES / *NO
.ASYNC	String	*YES / *NO
.LOCAL	String	*YES / *NO
.REMOTE	String	*YES / *NO
.OPTIONS	Struct	
.BULK-DATA	String	*YES / *NO
LOG	Struct	
.TRANS-F	String	*ON / *OFF / *FAILURE

Element	Type	Output
.FTAC	String	*ON / *REJECTED / *MODIFICATIONS
.ADM	String	*ON / *OFF / *FAILURE / *MODIFICATIONS
MAX-REQ-LIFETIME	String	*UNLIMITED / max-request-lifetime
SNMP-TRAPS	Struct	
.SUBSYSTEM-STATE	String	*OFF / *ON
.FT-STATE	String	*OFF / *ON
.PARTNER-STATE	String	*OFF / *ON
.PARTNER-UNREACHABLE	String	*OFF / *ON
.REQUEST-QUEUE-STATE	String	*OFF / *ON
.TRANSFER-SUCCESS	String	*OFF / *ON
.TRANSFER-FAILURE	String	*OFF / *ON
CONSOLE-TRAPS[1]	String	*OFF / *ON
CONS-TRAPS	Struct	
.SUBSYSTEM-STATE	String	*OFF / *ON
.FT-STATE	String	*OFF / *ON
.PARTNER-STATE	String	*OFF / *ON
.PARTNER-UNREACHABLE	String	*OFF / *ON
.REQUEST-QUEUE-STATE	String	*OFF / *ON
.TRANSFER-SUCCESS	String	*OFF / *ON
.TRANSFER-FAILURE	String	*OFF / *ON
ADM-TRAPS	Struct	
.DESTINATION	Struct	
. PARTNER	String	Value
.SELECTION	Struct	
.FT-STATE	String	*OFF / *ON
.PARTNER-STATE	String	*OFF / *ON
.PARTNER-UNREACHABLE	String	*OFF / *ON
.REQUEST-QUEUE-STATE	String	*OFF / *ON
.TRANSFER-SUCCESS	String	*OFF / *ON
.TRANSFER-FAILURE	String	*OFF / *ON
ADM-CONN-LIM	Integer	Value
HOST-NAME	String	Name of the BCAM host
IDENTIFICATION	String	Identification of the local openFT instance

Element	Type	Output
DYNAMIC-PARTNERS	String	*ON / *OFF
KEY-LEN	Integer	Value
STARTED	String	*YES / *NO
OPENFT-APPLICATION	String	*STD / Value
FTAM-APPLICATION	String	*STD / Value
FTP-PORT	String	*NONE / Value
OPENFT-STD	String	*STD / Value
ADM-PORT	String	Value
OPENFT-APPL-STATE	String	*DISABLED / *ACTIVE / *INACTIVE
FTAM-APPL-STATE	String	*NAVAIL / *DISABLED / *ACTIVE / *INACTIVE
FTP-STATE	String	*NAVAIL / *DISABLED / *ACTIVE / *INACTIVE
ADM-STATE	String	*DISABLED / *ACTIVE / *INACTIVE
MONITORING	Struct	
.STATE	String	*ON / *OFF
.PARTNER-SEL	Struct	
.OPENFT	String	*YES / *NO
.FTAM	String	*YES / *NO
.FTP	String	*YES / *NO
.REQUEST-SEL	Struct	
.SYNC	String	*YES / *NO
.ASYNC	String	*YES / *NO
.LOCAL	String	*YES / *NO
.REMOTE	String	*YES / *NO
ACTIVE-APPLICATIONS	Struct	
.OPENFT	String	*ON / *OFF
.FTP	String	*ON / *OFF
.ADM	String	*ON / *OFF
ENC-MAND	Struct	
.IN	String	*YES / *NO
.OUT	String	*YES / *NO
DEL-LOG	Struct	
.STATE	String	*ON / *OFF
.RETENTION	Integer	Value

Element	Type	Output
.REPEAT	String	*DAILY / *WEEKLY / *MONTHLY
.DAY	Integer	Value
.TIME	String	hh:mm:ss
ENC-MAND	Struct	
.IN	String	*YES / *NO
.OUT	String	*YES / *NO

[1] Now only support for reasons of compatibility. The value is only set if all the console traps are activated (*ON) or if all the console traps are deactivated (*OFF).

Meaning of the output of the OPS variables

Only the OPENTFT-STD variable is described below. The meanings of the other variables correspond to the associated output parameters of SHOW-FT-OPTIONS, see page 323.

OPENFT-STD
Port number used to address openFT partners if these are addressed via their host names without any port number specification.
*STD means that the default port number 1100 is used.
The value can be modified using the OPENFT-STF operand in the MODIFY-FT-OPTIONS command
Default setting following installation: *STD

4.27.1 Description of the output

Example

Default of the SHOW-FT-OPTIONS command, i.e. the operating parameters have not been modified since installation.

```
/SHOW-FT_OPTIONS
STARTED PROC-LIM CONN-LIM ADM-CLIM RQ-LIM MAX-RQ-LIFE TU-SIZE KEY-LEN
  YES       2       16        8     2000      30       65535   2048
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
*STD             *STD              21                 11000
ACTIVE           ACTIVE            ACTIVE             ACTIVE
HOST-NAME         IDENTIFICATION
*NONE             BS2FTPC

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
SNMP    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: 8

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

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 SHOW-FT-PARTNERS 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
Port number of the local FTAM server, where necessary supplemented by the transport selector, session selector and presentation selector. *STD means that the default value is used (port number 4800 and $FTAM as the transport selector).
Default setting following installation: *STD

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 BCAM host. The default value is *NONE, i.e. the real BCAM host is used.
FTMODOPTDefault setting following installation: *NONE

IDENTIFICATION
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: Name of the local BCAM host

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, SNMP 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.

SNMP
Setting for SNMP traps. .

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

4.28 SHOW-FT-PARTNERS Display partner systems

Note on usage

User group: FT user and FT administrator

Alias name: FTSHWPTN

Functional description

The SHOW-FT-PARTNERS command is used to obtain the following information on partner systems included in the partner list :

- 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 SHOW-FT-RANGE 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 or to an OPS variable: also the time of the last access and the authentication level.

SHOW-FT-PARTNERS 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 SHOW-FT-PARTNERS 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

SHOW-FT-PARTNERS / **FTSHWPTN**
PARTNER = ***ALL** / <text 1..200 with-low> **,OUTPUT** = ***SYSOUT**(...) / ***SYSLST**(...) ***SYSOUT**(...) / ***SYSLST**(...) **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 84

OUTPUT =
Output medium.

OUTPUT = *SYSOUT(...)
Output is sent to SYSOUT.

OUTPUT = *SYSLST(...)
Output is sent to SYSLST.

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 155).

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.
If this output is redirected to a file by using the SYSFILE command, it should be noted that the BS2000 Sysfile Management inserts a blank (i.e., a linefeed character) before each line. The first column of the file must hence be stripped before the procedure generated by this method can be called. We therefore recommend that you use the START-OPENFTPART command, which performs this task for the user.

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 334.

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.

Command return codes

(SC2)	SC1	Maincode	Meaning
0	0	CMD0001	No partner available for the selection criteria.
83	32	CMD0221	Internal error.
88	32	CMD0221	Error during OPS generation.
35	64	FTR1035	The user is not authorized to use this command.
45	64	FTR1045	No partner found in partner list.
2	0	FTR2225	Information output cancelled.

SC1/2 = Subcode 1/2 in decimal notation

For additional information, see section “Command return codes” on page 153

OPS variables

The following table shows the OPS variables for the command SHOW-FT-PARTNERS.

Element	Type	Output
PARTNER-NAME	String	
STA	String	*ACTIVE / *INACTIVE / *NO-CONN / *LOC-UNKNOWN /*REM-UNKNOWN / *ACTIVE (AUTO-DEACTIVATE) / *INACTIVE (BY- AUTOMATIC-DEACTIVATION) / *LOC-AUTH-FAIL / *REM-AUTH-FAIL / *DATA-INTEGRITY-ERROR / *NO-KEY / *ID-REJ
SEC-LEV	Integer/ String	VALUE / *B-P-ADDR
TRACE [1]	String	*ON / *OFF / *BY-FT-OPT
LOC	Integer	Value
REM	Integer	Value
PARTNER-ADDR	String	Value
PRIO	String	*NORM / *HIGH / *LOW
AUTHENTICATION-LEVEL	Integer	1 / 2 / empty
LAST-ACCESS-DATE	String	Value / empty
ADDR-TYPE [2]	String	*OPENFT / *PRESENTATION / *TCP-IP
OPENFT-ADDR [3]	Struct	
.PROCESSOR	String	Value
.ENTITY	String	Value
.NETWORK-ADDR	String	Value
.TRANS-SEL	String	Value
.PORT	String	port number

Element	Type	Output
.PARTNER-CHECK	String	*FTOPT / *STD / *TRANSP-ADDR / *AUTH
.AUTH-MAND	String	*YES / *NO
.IDENTIFICATION	String	Value
.SESSION-ROUTING	String	*ID or empty
PRESENTATION-ADDR [4]	Struct	
.NETWORK-ADDR	String	Value
.TRANSPORT-SEL	String	Value
.SESSION-SEL	String	Value
.PRESENTATION-SEL	String	Value
.PORT	String	Value
TCP-IP-ADDR [5]	Struct	
.PORT	String	Value
ADM-ADDR [6]	Struct	
.PROCESSOR	String	Value
.ENTITY	String	Value
.NETWORK-ADDR	String	Value
.TRANS-SEL	String	Value
.PORT	String	Port number
.PARTNER-CHECK	String	*FTOPT / *STD / *TRANSP-ADDR /*AUTH
.AUTH-MAND	String	*YES / *NO
.IDENTIFICATION	String	Value
.SESSION-ROUTING	String	*ID / empty
INBOUND-STATE	String	*ACTIVE / *INACTIVE
REQ-PROC	String	*STD / *SERIAL

[1] TRACE is only displayed for openFT partners.

[2] Only the address structure corresponding to the ADDR-TYPE element is displayed.

[3] Only applies to openFT partners.

[4] Only applies to FTAM partners.

[5] Only applies to FTP partners.

[6] Only applies to ADM partners

Example

Request information on all remote systems entered in the partner list:

Short output:

```
/SHOW-FT-PARTNERS 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
FOREIGN   ACT     10      NORM FTOPT      0       0       ftam://PC3:102.ftam.
                                                          ftam.ftam
FTAMPC    ACT     30      NORM FTOPT      0       0       ftam://PC2:.$ftam
FTAMUX    ACT     30      NORM FTOPT      0       0       ftam://UNIX3
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:

```
/SHOW-FT-PARTNERS 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
HUGO      ACT   STD       NORM FTOPT    0    0 FTOPT HUGO
          ACT   STD                                          %.HUGO.$FJAM
HOSTABS2  ACT   B-P-ATTR  NORM FTOPT    0    0 FTOPT HOSTABS2
          ACT   STD                                          HOSTABS2.FUJI.NET
HOSTBBS2  ACT   STD       NORM FTOPT    0    0 FTOPT HOSTBBS2
          ACT   STD                                          HOSTBBS2.CLOUD.NET
FOREIGN   ACT     10      LOW  FTOPT    0    0       ftam://PC3:102.ftam.
          ACT   STD                                          ftam.ftam
FTAMPC    ACT     30      NORM FTOPT    0    0       ftam://PC2:.$ftam
          ACT   STD                                          ftamw.ftam2
FTAMUX    ACT     30      NORM FTOPT    0    0       ftam://UNIX3
          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 CREATE-FT-KEY-SET .

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 MODIFY-FT-OPTIONS 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 MODIFY-FT-PARTNER 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 84.

IDENTIFICATION
Instance ID of the partner (also see the ADD-FT-PARTNER command in the System Administrator Guide).

ROUTING
SESSION-ROUTING-INFO of the partner, where required (also see the ADD-FT-PARTNER 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.

4.29 SHOW-FT-PROFILE 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 SHOW-FT-PROFILE, 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 SHOW-FT-PROFILE to access defined passwords or transfer admissions defined in the profile! If a transfer admission is forgotten, a new one must be specified using MODIFY-FT-PROFILE.

Format

SHOW-FT-PROFILE

NAME = ***ALL** / <alphanum-name 1..8> / *STD

,SELECT-PARAMETER = ***OWN** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

TRANSFER-**ADM**ISSION = ***ALL** / ***NOT-SPEC**IFIED / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> / ***SEC**RET

,OWNER-**ID**ENTIFICATION = ***OWN** / <name 1..8>

,INFORMATION = ***ONLY**-NAMES / ***ALL**

,OUTPUT = ***SYSOUT**(...) / ***SYSLST**(...)

***SYSOUT**(...) / ***SYSLST**(...)

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.

TRANSFER-ADMISSION = *SECRET
The system prompts you to enter the transfer admission. However, this does not appear on the screen.

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.

OUTPUT =
Output medium for the information.

OUTPUT = *SYSOUT(...)
Output is sent to SYSOUT.

OUTPUT = *SYSLST(...)
Output is sent to SYSLST.

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 155).

Command return codes

(SC2)	SC1	Maincode	Meaning
0	64	FTC0052	The information output was interrupted.
0	64	FTC0053	No FT profile exists which meets the specified criteria.
0	0	FTC0054	No information exists for the specified criteria.
0	64	FTC0153	The owner identification entered is not the user's own ID.
0	64	FTC0171	The profile entered does not exist.
0	64	FTC0255	A system error occurred.

SC1/2 = Subcode 1/2 in decimal notation
For additional information, see section "Command return codes" on page 153

OPS variables

The following table shows the OPS variables of the SHOW-FT-PROFILE command with the operand INF=*ALL. The underlined values apply to the output with INF=*ONLY-NAMES.

Element	Type	Output
PROF-NAME	String	
PRIV	String	*YES / *NO
TRANS-ADM	String	*NSPEC / *SECRET
DUPLICATED	String	*YES / *NO
LOCKED-BY	Struct	
.IMPORT	String	*YES / *NO
.ADM	String	*YES / *NO
.USER	String	*YES / *NO
EXPIRED	String	*YES / *NO
USER-ADM	Struct	
.USER-ID	String	User-ID
.ACC	String	Account number / *FIRST / *NSPEC / *NONE / *NRES
.PASSWORD	String	*OWN / *NSPEC / *NONE / *YES
EXP-DATE	String	yyyy-mm-dd / *NRES
USAGE	String	*PUBLIC / *PRIVATE / *NSPEC
IGNORE	Struct	
.OBS	String	*YES / *NO
.OBR	String	*YES / *NO
.IBS	String	*YES / *NO

Element	Type	Output
.IBR	String	*YES / *NO
.IBP	String	*YES / *NO
.IBF	String	*YES / *NO
INITIATOR	String	*LOC / *REM / *NRES
TRANS-DIR	String	*FROM / *TO / *NRES
MAX-PART-LEV	String	Maximum security level / *NRES
PARTNERS	Array (1-50)	One or several partners / *NRES
FILE-NAME	String	File name / *NRES
LIBRARY	String	*YES / *NO / *NRES / Library
FILE-NAME-PREFIX	String	*YES / *NO
ELEM	Struct	
.NAME	String	Name / *NRES / *NONE
.PREFIX	String	*YES / *NO
.VERSION	String	Version / *STD / *NONE / *NRES
.TYPE	String	Type / *NRES / *NONE
FILE-PASSWORD	String	*YES / *NRES / *NONE
WRITE	String	*NEW / *EXT / *REPL / *NRES
PROC-ADM	Struct	
.USER-ID	String	User-ID / *NRES / *SAME
.ACC	String	Account number / *NRES / *SAME / *NONE
.PASSWORD	String	*NONE / *YES / *NRES / *SAME
SUCC	Struct	
.PROC	String	Commands / *NONE / *NRES / *EXPANSION
.PREFIX	String	Prefix / *NONE
.SUFFIX	String	Suffix / *NONE
FAIL	Struct	
.PROC	String	Commands / *NONE / *NRES / *EXPANSION
.PREFIX	String	Prefix / *NONE
.SUFFIX	String	Suffix / *NONE
TRANS-FILE	String	*ALLOWED / *NOT-ALLOWED
MOD-FILE-ATTR	String	*ALLOWED / *NOT-ALLOWED
READ-DIR	String	*ALLOWED / *NOT-ALLOWED
FILE-PRO^C	String	*ALLOWED / *NOT-ALLOWED

Element	Type	Output
ACC-ADM	String	*ALLOWED / *NOT-ALLOWED
REM-ADM	String	*ALLOWED / *NOT-ALLOWED
ADM-TRAP-LOG	String	*ALLOWED / *NOT-ALLOWED
TEXT	String	Text / *NONE
DATA-ENC	String	*YES / *NO / *NRES
LAST-MOD	Struct	
.DATE	String	yyyy-mm-dd / *NONE
.TIME	String	hh:mm:ss / *NONE

Example

The user STEVEN wishes to view his admission profile UMSAWARE with the command SHOW-FT-PROFILE to determine if the profile might endanger data protection:

```
/SHOW-FT-PROFILE␣NAME=UMSAWARE, -
INFORMATION=*ALL
```

Short form:

```
/SHOW-FT-PROF␣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 CREATE-FT-PROFILE 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 CREATE-FT-PROFILE 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 MODIFY-FT-PROFILE 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.

4.30 SHOW-FT-RANGE 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 SHOW-FT-RANGE 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 SHOW-FT-ADMISSION-SET to obtain information on your admission set (see page 277).

Format

SHOW-FT-RANGE

USER-IDENTIFICATION = ***OWN** / <name 1..8>

,SELECT-PARAMETER = ***ALL** / ***PAR**AMETERS(...)

***PARA**METERS(...)

PARTNER = ***ALL** / <text 1..200 with-low>

,OUTPUT = ***SYSOUT**(...) / ***SYSLST**(...)

***SYSOUT**(...) / ***SYSLST**(...)

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.
– if no authorization exists for the partner system, FTC0170 is displayed.
For additional information to partner addresses, see section “Defining the partner computer” on page 84.

OUTPUT =
Output medium for the partner system listing.

OUTPUT = *SYSOUT(...)
The list is output on SYSOUT.

OUTPUT = *SYSLST(...)
The list is output on SYSLST.

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 155).

Example

Steven Miller would like to find out about the security level of the computer BUYDACK. To do this, he uses the following command:

```
/SHOW-FT-RANGE␣SELECT-PARAMETER=(PARTNER=BUYDACK)
```

Short form:

```
/SHOW-FT-RANGE␣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.

Command return codes

(SC2)	SC1	Maincode	Meaning
0	64	FTC0052	The output of information was interrupted.
0	0	FTC0054	There is no information which meets the specified criteria.
0	64	FTC0070	The command cannot be executed on the basis of inadequate operating resources.
0	64	FTC0152	The user ID entered is not the user's own ID.
0	64	FTC0170	The partner entered is unknown within the partner systems possible for this user.
0	64	FTC0255	A system error occurred.

SC1/2 = Subcode 1/2 in decimal notation
For additional information, see section "Command return codes" on page 153

OPS variables

Element	Type	Output
SEC-LEV	Integer	Security level
PARTNER-NAME	String	Partner name

4.31 SHOW-REMOTE-FILE-ATTRIBUTES Display remote file attributes

Note on usage

User group: FT user

Alias name: FTSHW

Functional description

With the SHOW-REMOTE-FILE-ATTRIBUTES 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

SHOW-REMOTE-**FILE-ATTR**IBUTES / **FTSHW**

PARTNER = <text 1..200 with-low>

,**FILE** = **_*NOT-SPECIFIED_** / <filename 1..54> / <c-string 1..512 with-low> / ***DIR**ECTORY(...)

***DIR**ECTORY(...)
NAME = **_*NOT-SPECIFIED_** / <c-string 1..512 with-low> / <partial-filename 2..53>

,**PASS**WORD = **_*NONE_** / <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128> / ***SECRET**

,**TRANS**FER-**ADM**ISSION = **_*NONE_** / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> / ***SECRET** / ***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> / ***SECRET**

,**INF**ORMATION = **_*STD_** / ***ALL-ATTR**IBUTES / ***NAMES-ONLY**

,OUTPUT = **_*SYSOUT_**(...) / ***SYSLST**(...)

*SYSOUT(...) / ***SYSLST**(...)
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 84.

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..54> / <c-string 1..512 with-low>
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> / <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.

PASSWORD = *SECRET
The system prompts you to enter the password. However, the input is not displayed on the screen.

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 = *SECRET
The system prompts you to enter the transfer admission. However, the input is not displayed on the screen.

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.

PASSWORD = *SECRET
The system prompts you to enter the password. However, the input is not displayed on the screen.

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 = *SYSOUT(...)
Output is send to SYSOUT.

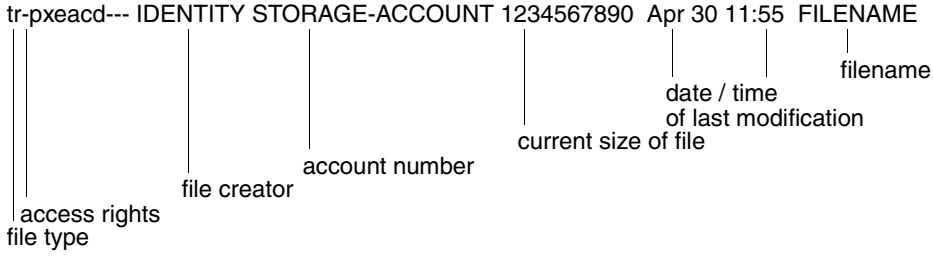
OUTPUT = *SYSLST(...)
Output is send to SYSLST.

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 155).

4.31.1 Description of the output

Default output



Long output form

```
FILENAME=REMDATEI
STORAGE-ACCOUNT=a1234ft
CRE   MAIER DATE=Nov 19 11:55
MOD   DATE=Apr 28 15:54
REA   DATE=Apr 30 09:01
CHARACTERSET=g   RECORD-FORMAT=v   RECORD=SIZE=1200
FILE-AVAILIBILITY=I ACCESS=RIGHTS=r-pxeacd---    FILESIZE=123456
MAX-FILESIZE=1234567 LEGAL-QUALIFICATION=STRING
```

4.31.2 Description of output fields

CHARACTERSET – File type

This specifies the file type. This field can have the following values in the default output:

t file contains text
b file contains binary data
d directory
* no information on data structure available

The long output form is as follows:

BINARY-FILE	binary file
DIRECTORY	directory
CHARACTERSET	text file *

For text files (CHARACTERSET=), the character set of the characters of the text file is indicated. The field can have the following values:

g graphic string *)
The file can contain characters from the G0 set of ISO646 or from the G0 set of ISO8859-1 and the G1 set of ISO8859-1.
c general string *)
The file can contain characters from the C0 set of ISO646 and either from the G0 set of ISO646 or ISO8859-1 and from the G1 set of ISO8859-1.
i IA5 string *)
The file can contain characters from the C0 set and G0 set of ISO646.
v visible string *)
The file can contain characters from the G0 set of ISO646.

Values marked with * are only relevant for FTAM.

ACCESS-RIGHTS – Access rights

Contains information about how a file can be accessed. The field can contain the following values:

r file can be sent.
i data units can be entered. *)
p file can be overwritten.
x file can be expanded, i.e. data can be added to the file.
e data units can be deleted from the file.
a file attributes can be read.
c file attributes can be modified.
d file can be deleted.
t access direction forwards (traversal) *)
v access direction backwards (reverse traversal) *
r random access *)

Values marked with * are only relevant for FTAM.

File creator

Identity of the person who created the file. In BS2000, the information consists of the user ID under which the file is created. In Unix systems, this value generally indicates the owner of the file. The field can be up to 32 characters long.

STORAGE-ACCOUNT – Account number

FTAM-specific value. It contains the account number for which the costs are calculated for saving the file in the remote system.

FILE-SIZE – Current file size in bytes

This contains the current size of the file in bytes. The output is only as accurate as the information returned by the partner system. Since the files can be set up differently in different systems, files of the same size may have different values in this field, depending on the system. Some memories assign multiples of basic units, or blocks, for file storage. Therefore, it is important that the value indicated here not be taken as the actual size, but rather be treated as a guideline.

In the case of LMS library members, it is particularly important to note that the displayed size is generally smaller than the file resulting from the transfer since library members can always be stored in compressed form (see the section “Structure of a library” in the LMS manual).

Date and time of last modification of file contents

This contains information on the time of the last modification. For modifications which have been made in the past six months, the output takes the form *MonthDayTime* (e.g. Apr 25 15:13); for modifications which were made over six months ago, the form is *MonthDayYear* (e.g. Apr 30 2012).

FILENAME – File name

This contains the file name.

The following values are only included in the long output form:

CRE, MOD, REA, ATM – Type of last file use

This contains information about how the file was last accessed. The following actions can be displayed:

CRE file creation
MOD modification of file contents (overwriting, expansion)
REA * read file (send)
ATM * modification of file attributes
Values marked with * are only relevant for FTAM.

It must be noted that it is left to the remote system to determine what information will be returned. Therefore, the lines with the information on the file use may look different for each partner system and contain different information. Generally, the information about the creation of the file is supplied, if nothing else.

Information about the modification of file contents or file attributes, about sending a file, or about when the file was last used may not be included.

Name of last file user

Identity of the last user who accessed the file in a particular way.

RECORD-FORMAT – Record format

This contains the format of the transferred records. Possible values are:

v variable length records
f uniform length records
u no record structure or variable length records, all of which are terminated by CRLF (carriage return line feed) for transfer.

RECORD-SIZE – Maximum record length

This contains the maximum length of the transferred records.

FILE-AVAILABILITY – File availability

FTAM-specific value. The field can have the following values:

i the file is available immediately (immediate).
d the file is not available immediately (deferred).
The meaning of the word “deferred” is determined by the partner system.

MAX-FILESIZE – Possible file size in bytes

FTAM-specific value. It contains the possible file size. The specification is only as accurate as the information sent by the partner system. Since the files can be set up differently in different systems, files of the same size may have different values in this field, depending on the system. Therefore, it is important that the value indicated here not be taken as the actual size, but rather be treated as a guideline.

LEGAL-QUALIFICATION – Legal qualification

FTAM-specific value which contains the legal qualifications for a file (in lieu of a copyright).

Command return codes

(SC2)	SC1	Maincode	Meaning
0	130	FTR0108	Request rejected. Remote system not accessible.
33	32	CMD0221	Request rejected. Internal error.
36	32	CMD0221	Request rejected. Request data inconsistent.
83	32	CMD0221	Internal error.
37	64	FTR2037	File is read only.
43	64	FTR2043	Access denied.
120	64	FTR2120	Remote system unknown in the local system
123	64	FTR2123	Request rejected. OSS call error. The command was not executed because the session instance detected a communication error.
148	64	FTR2148	Request rejected. Remote system: Transfer of file generation groups not supported.
152	64	FTR2152	Request rejected. Remote system: No file name specified.
153	64	FTR2153	Request rejected. Remote system: Invalid management password.
155	64	FTR2155	Request rejected. Remote system: File not found.
169	64	FTR2169	Request rejected. Remote system: Transfer admission invalid. Transfer admission incorrect or insufficient FTAC authorizations.
170	64	FTR2170	Request rejected. Remote system: Function not supported.

SC1/2 = Subcodes 1/2 in decimal notation

For additional information, see section “Command return codes” on page 153

OPS variables

The following table shows the OPS variables for the command SHOW-REMOTE-FILE-ATTRIBUTES with the operand INF=*ALL-ATTRIBUTES, the underlined values are only valid for the output with the operand INF=*STD. For the operand input INF=*NAMES-ONLY only the element F-NAME (string) will be output.

Element	Type	Output
F-NAME	String	
STOR-ACCOUNT	String	
CRE	Struct	
.USER	String	
.DATE	String	yyyy-mm-dd
.TIME	String	hh:mm:ss
MODIFY	Struct	
.USER	String	
.DATE	String	yyyy-mm-dd
.TIME	String	hh:mm:ss
READ	String	
.USER	String	
.DATE	String	yyyy-mm-dd
.TIME	String	hh:mm:ss
MOD-ATTR	String	
.USER	String	
.DATE	String	yyyy-mm-dd
.TIME	String	hh:mm:ss
DATA-TYPE	String	*CHAR / *BINARY / *DIR / *NO-INFO
CHAR-SET [1]	String	*GRAPHIC / *GENERAL / *IA5 / *VISIBLE
REC-FORM	String	*VAR / *FIXED / *UNDEF
REC-SIZE	Integer	
F-AVAIL	String	*IMMED / *DEFERRED
ACCESS-RIGHTS	Struct	
.READ-F	String	*YES / *NO
.INS-DATA-UNIT	String	*YES / *NO
.REPLACE-F	String	*YES / *NO
.EXTEND-F	String	*YES / *NO

Element	Type	Output
.ERASE-DATA-UNIT	String	*YES / *NO
.READ-ATTR	String	*YES / *NO
.MOD-ATTR	String	*YES / *NO
.DEL-F	String	*YES / *NO
.TRAVERSAL	String	*YES / *NO
.REV-TRAVERSAL	String	*YES / *NO
.RANDOM	String	*YES / *NO
F-SIZE	Integer	
MAX-F-SIZE	Integer	
LEGAL-QUALIFICATION	String	
CCS-NAME	String	Value

[1] The element CHAR-SET is only assigned if DATA-TYPE=*CHAR.

Examples

1. You wish to output the standard information on the REMFILE in the BS2000 computer with the symbolic name HUGO under the login JIM with account number A1234FT and password C'PWD'.

```
/SHOW-REMOTE-FILE-ATTRIBUTES␣PARTNER = HUGO,FILE-NAME = REMFILE,     -
/                           TRANSFER-ADMISSION = (JIM,A1234FT,C'PWD')

%*r-pxeacd--- JIM               1234567890 Apr 30 11:55 REMFILE
```

   Short form:

```
/SH-REM-FI-ATT HUGO,REMFILE,,(JIM,A1234FT,'PWD')
```

2. You wish to output detailed information on the attributes of the REMFILE file on the FTAM partner system with the symbolic name HUGO under the login JIM with the account number A1234FT and password C'PWD':

```
/SHOW-REMOTE-FILE-ATTRIBUTES PARTNER = HUGO,FILE-NAME = REMFILE,        -
/                           TRANSFER-ADMISSION = (JIM,A1234FT,C'PWD'),  -
/                           INFORMATION = *ALL-ATTRIBUTES

%FILENAME=REMFILE
%CRE   JIM
%MOD   DATE=Apr 28 15:54
%RECORD-FORMAT=v
%ACCESS=RIGHTS=r-pxeacd---    FILESIZE=123456
```

   Short form:

```
/SH-REM-FI-ATT HUGO,REMFILE,,(JIM,A1234FT,'PWD'),*ALL
```

3. You wish to obtain comprehensive information concerning the attributes of the POSIX files in the remote system *compute1*. The transfer admission for the remote system is *number13*.

   a) The output relates to the POSIX directory file:

```
/SHOW-REMOTE-FILE-ATTRIBUTES compute1,*DIR('./file'),,number13

%*r-pxeacd--- AXL                    2416640 Feb 13 10:18 FUTURE-E.28
%*r-pxeacd--- AXL                   26365952 Feb 09 09:36 FUTURE-E.26
%tr-pxeacd--- AXL                       2048 Feb 16 15:24 TEST
%*r-pxeacd--- AXL                     524288 Jan 26 10:11 NSTD
```

   b) Output is requested for the POSIX file named file.test:

```
/SHOW-REMOTE-FILE-ATTRIBUTES compute1,‘./file.test‘,,number13

%*r-pxeacd--- axl                        2048 Feb 17 15:50 ./file.test
```

   c) The output is requested for the entire directory:

```
/SHOW-REMOTE-FILE-ATTRIBUTES compute1,*DIR(‘./.‘),,number13

%*r-pxeacd--- axl                           0 Feb 17 16:55 Z1.FUTURE-F.780
%*r-pxeacd--- axl                           0 Feb 22 10:05 Z1.FUTURE-F.1060
%*r-pxeacd--- axl                      261389 Feb 17 11:32 F-703-0BUP
%*r-pxeacd--- axl                           0 Feb 20 10:39 SCRATCH
%*r-pxeacd--- axl                       21511 Feb 22 10:46 Z3.FUTURE-F.107
%*r-pxeacd--- axl                      199355 Feb 17 11:20 Z1.FUTURE-F.739
%*r-pxeacd--- axl                       19899 Feb 17 10:23 Z1.FUTURE-F.693
%*r-pxeacd--- axl                           0 Feb 15 20:40 F-291-0BM8
%*r-pxeacd--- axl                           0 Feb 15 16:28 F-172-0BM7
%*r-pxeacd--- axl                        2048 Feb 17 15:50 file.test
```

4. All files that start with FILE. are to be displayed:

```
/SHOW-REMOTE-FILE-ATTRIBUTES compute1,*DIR(file.),,number13

%*r-pxeacd--- AXL                    2416640 Feb 13 10:10 INCOMING
%*r-pxeacd--- AXL                   26365952 Feb 09 09:36 OUTGOING
```

5. All characteristics of the file FILE.TEST should be displayed in CSV format at the system PARTBS2:

```
/SH-REM-FI PARTBS2,FILE.TEST,,transadm,OUT=*SYSOUT(*CSV),INF=*ALL

FileName;StorageAccount;CreIdentity;CreTime;ModIdentity;ModTime;
ReaIdentity;ReaTime;AtmIdentity;AtmTime;FileType;CharSet;RecFormat;
RecSize;FileAvail;AccessRights;FileSize;MaxFileSize;LegalQualif;CcsName
"FILE.TEST";*NSPEC;"UID";*NSPEC;*NSPEC;2012-07-05 14:43:13;*NSPEC;
*NSPEC;*NSPEC;*NSPEC;*NSPEC;*NSPEC;*VAR;*NSPEC;*NSPEC;r-pxeacd---;2048;
*NSPEC;*NSPEC;*NSPEC
```

6. The properites of the PLAM library TESTLIB are to be displayed.

   PLAM libraries are addressed as directory structures:
   Library/Type/Member or Library/Type/Member()/Member(Version)

   a) To list all types of a library:

```
/SHOW-REM-FIL-ATTR PCVQ7C2,*DIR('TESTLIB'),,TRANSADM

%dr-pxeacd--- TSOS                    Jul 25 13:06 s
%dr-pxeacd--- TSOS                    Jul 25 13:06 x
```

   b) To list all members of a type:

```
/SHOW-REM-FIL-ATTR PCVQ7C2,*DIR('TESTLIB/S'),,TRANSADM
*r-pxeacd--- TSOS              2048  Jul 25 13:05 test
dr-pxeacd--- TSOS                    Jul 25 13:06 test()
*r-pxeacd--- TSOS              2048  Jul 25 13:06 test.outfile
```

   Note:
   All members of a type are listed as files in the directory of this type. When transfer takes place, the highest version (or Version @) is supplied. If more than one version of a member exists, a Member() directory is also displayed in which the other versions of the member are contained.

   c) To list all additional versions of the member test:

```
/SHOW-REM-FIL-ATTR PCVQ7C2,*DIR('TESTLIB/S/test()'),,TRANSADM
*r-pxeacd--- TSOS              2048  Jul 25 13:06 test(12)
```

   d) To display the highest version of the member test:

```
/SHOW-REM-FIL-ATTR PCVQ7C2,'TESTLIB/S/test',,TRANSADM
*r-pxeacd--- TSOS              2048  Jul 25 13:05 testlib/s/test
```

   e) To display Version 12 of the member test:

```
/SHOW-REM-FIL-ATTR PCVQ7C2,'TESTLIB/S/test()/test(12)',,TRANSADM

*r-pxeacd--- TSOS              2048  Jul 25 13:06 testlib/s/test()/test(12)
```

4.32 TRANSFER-FILE Transfer file asynchronously

Note on usage

User group: FT user

Alias names: TFF / NCOPY / FTACOPY

Functional description

The TRANSFER-FILE command can be used to transfer a file or a library member to or from a partner system.

The local system is regarded as the system in which the command is issued, or in this case, the BS2000 computer. The partner system is designated as the remote system.

4.32.1 Introduction to the TRANSFER-FILE 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 378).

The remaining “optional” operands (see page 395) are used to define the other characteristics of the file transfer, such as compressed or encrypted transfer or the starting time for the transfer.

4.32.1.1 The shortest form of the command

The mandatory parameters for the TRANSFER-FILE 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)

A file transfer can be effected using these three parameters alone, if:
– the user ID for the data transfer is the same in both systems
– the user ID is not password-protected
– the FT user issues the command under this user ID
– the file name is the same in both systems,
– the send and receive files are not password-protected

An example can be found on page 403.

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 363ff.

4.32.1.2 How to find out if the file transfer request has been executed

The command SHOW-FILE-TRANSFER 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 TRANSFER-FILE 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,
– use a job variable to monitor the FT request (not for requests with FTAM partners).

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.

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 99.

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.

4.32.2 Full form of the TRANSFER-FILE command

Format

(part 1 of 5)

TRANSFER-**FILE** / **TFF** / **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..54> / <c-string 1..512 with-low> /
***LIB**RARY-**ELEM**ENT(...) / ***POS**IX(NAME=<posix-pathname 1..510>)

***LIB**RARY-**ELEM**ENT(...)

LIBRARY = ***NOT-SPECIFIED** / <filename 1..54>

,**ELEM**ENT = ***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>

,**PASS**WORD = ***NONE** / <c-string 1..4> / <x-string 1..8> /
<integer -2147483648..2147483647> / ***SECRET**

,**TRANS**FER-**ADM**ISSION = ***SAME** / <alphanum-name 8..32> / <x-string 15..64> /
<c-string 8..32 with-low> / ***SECRET** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

USER-IDENTIFICATION = <name 1..8>

,**ACCOUNT** = ***NONE** / <alphanum-name 1..8>

,**PASS**WORD = ***NONE** / <c-string 1..8> / <c-string 9..32> /
<x-string 1..16> / ***SECRET**

,**PROC**ESSING-**ADM**ISSION = ***SAME** / ***NOT-SPECIFIED** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

USER-IDENTIFICATION = <name 1..8>

,**ACCOUNT** = ***NONE** / <alphanum-name 1..8>

,**PASS**WORD = ***NONE** / <c-string 1..8> / <c-string 9..32> / <x-string 1..16> / ***SECRET**

,**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(...)

OUTPUT = ***SYSLST** / ***LISTFILE**

,CONDITION = ***ANY** / ***ON-FAILURE-ONLY**

,**MONJV** = ***NONE** / <filename 1..54>

,**JV-PASS**WORD = ***NONE** / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647> /
***SECRET**

,**CODED-CHAR**ACTER-**SET** = ***STD** / <name 1..8>

(part 2 of 5)

```
,REMOTE-PARAMETER = *BS2000(...) / *MSP(...) / *ANY(...)
  *BS2000(...)
    FILE-NAME = *SAME / *NOT-SPECIFIED  / <filename 1..54> / <c-string 1..512 with-low> /
                  *LIBRARY-ELEMENT(...) / *POSIX(NAME=<posix-pathname 1..510>)
      *LIBRARY-ELEMENT(...)
        LIBRARY = *SAME / *NOT-SPECIFIED / <filename 1..54>
        ,ELEMENT = *SAME / *NOT-SPECIFIED /
                     <filename 1..64 without-gen-vers>(...) / <composed-name 1..64 with-under>(...)
          <filename>(...) / <composed-name>(...)
            VERSION = *SAME / *STD / <text 1..24>
        ,TYPE = *SAME / *NOT-SPECIFIED / <name 1..8>
    ,PASSWORD = *SAME / *NONE / <c-string 1..4> / <x-string 1..8> /
                  <integer -2147483648..2147483647> / *SECRET
    ,TRANSFER-ADMISSION = *SAME / <alphanum-name 8..32> / <x-string 15..64> /
                            <c-string 8..32 with-low> / *SECRET / *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> / *SECRET
    ,PROCESSING-ADMISSION = *SAME / *NOT-SPECIFIED / *PARAMETERS(...)
      *PARAMETERS(...)
        USER-IDENTIFICATION = <c-string 1..8>
        ,ACCOUNT = *NONE / <alphanum-name 1..8>
        ,PASSWORD = *NONE / <c-string 1..8> / <c-string 9..32> / <x-string 1..16> / *SECRET
    ,SUCCESS-PROCESSING = *NONE / <c-string 1..1000 with-low>
    ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
    ,FILE-AVAILABILITY = *BY-RECEIVING-SYSTEM / *IMMEDIATE / *DEFERRED
    ,ACCESS-MODE = *BY-RECEIVING-SYSTEM / *PARAMETERS(...)
      *PARAMETERS(...)
        READ-FILE = *NO / *YES
        ,INSERT-DATA-UNIT = *NO / *YES
        ,REPLACE-FILE = *NO / *YES
        ,EXTEND-FILE = *NO / *YES
        ,ERASE-DATA-UNIT = *NO / *YES
        ,READ-ATTRIBUTES = *NO / *YES
        ,CHANGE-ATTRIBUTES = *NO / *YES
        ,DELETE-FILE = *NO / *YES
    ,CODED-CHARACTER-SET = *STD / <name 1..8>
```

(part 3 of 5)

```
*MSP(...)
    FILE-NAME = *NOT-SPECIFIED / <text 1..56>
   ,PASSWORD = *NONE / <alphanum-name 1..8> / *SECRET
   ,TRANSFER-ADMISSION = <alphanum-name 8..32> / <x-string 15..64> /  <c-string 8..32 with-low> /
                         *SECRET / *PARAMETERS(...)
      *PARAMETERS(...)
          USER-IDENTIFICATION = <name 1..7>
         ,ACCOUNT = *NONE /<text 1..43>
         ,PASSWORD = *NONE / <alphanum-name 1..8> / *SECRET
   ,PROCESSING-ADMISSION = *SAME / *NOT-SPECIFIED / *PARAMETERS(...)
      *PARAMETERS(...)
          USER-IDENTIFICATION = <name 1..7>
         ,ACCOUNT = *NONE /<text 1..43>
         ,PASSWORD = *NONE / <alphanum-name 1..8> / *SECRET
   ,SUCCESS-PROCESSING = *NONE /<c-string 1..1000 with-low>
   ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
   ,CODED-CHARACTER-SET = *STD / <name 1..8>
*ANY(...)
    FILE-NAME = *NOT-SPECIFIED / <filename 1..54> / <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> / *SECRET
   ,TRANSFER-ADMISSION = *NONE / <alphanum-name 8..32> / <x-string 15..64> /
                         <c-string 8..32 with-low> / *SECRET / *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> / *SECRET
   ,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> / *SECRET
```

(part 4 of 5)

,**SUCC**ESS-PROCESSING = **<u>*NONE</u>** /<c-string 1..1000 with-low>
,**FAIL**URE-PROCESSING = **<u>*NONE</u>** / <c-string 1..1000 with-low>
,**FILE-AVAIL**ABILITY = **<u>*BY-RECEIVING-SYSTEM</u>** / ***IMMED**IATE / ***DEFERRED**
,**STOR**AGE-**ACC**OUNT = **<u>*NONE</u>** / <c-string 1..40 with-low> / <text 1..40>
,**ACCESS-MODE** = **<u>*BY-RECEIVING-SYSTEM</u>** / ***PAR**AMETERS(...)
***PAR**AMETERS(...)
READ-FILE = **<u>*NO</u>** / ***Y**ES
,**INS**ERT-**DATA**-UNIT = **<u>*NO</u>** / ***Y**ES
,**REPLACE**-FILE = **<u>*NO</u>** / ***Y**ES
,**EXT**END-FILE = **<u>*NO</u>** / ***Y**ES
,**ERA**SE-**DATA**-UNIT = **<u>*NO</u>** / ***Y**ES
,**READ-ATTR**IBUTES = **<u>*NO</u>** / ***Y**ES
,**CHAN**GE-**ATTR**IBUTES = **<u>*NO</u>** / ***Y**ES
,**DEL**ETE-**FILE** = **<u>*NO</u>** / ***Y**ES
,**LEG**AL-**QUAL**IFICATION = **<u>*NONE</u>** / <c-string 1..80 with-low> / <text 1..80>
,**CRE**ATE-**PASS**WORD = **<u>*NONE</u>** / <c-string 1..64 with-low> / <x-string 1..128> / ***SECRET**
,**CODED-CHAR**ACTER-**SET** = **<u>*STD</u>** / <name 1..8>

,**COMP**RESS = **<u>*NONE</u>** / ***BYTE-REPET**ITION / ***ZIP**

,**WRITE**-MODE = **<u>*REPLACE</u>**-FILE / ***NEW-**FILE / ***EXT**END-FILE

,**DATA**-TYPE = **<u>*NOT-SPECIFIED</u>** / ***CHAR**ACTER (...) / ***BIN**ARY (...) / ***USER**

***CHAR**ACTER(...)
TRANSPARENT = **<u>*NO</u>** / ***YES**

***BIN**ARY(...)
TRANSPARENT = **<u>*NO</u>** / ***YES**

,**PRIO**RITY = **<u>*NORMAL</u>** / ***HIGH** / ***LOW**

,**START** = **<u>*SOON</u>** / ***EARL**IEST(...)

***EARL**IEST(...)
DATE = **<u>*TODAY</u>** / ***TOMOR**ROW / <date 8..10>
,**TIME** = **<u>00:00</u>** / <time 1..8>

,**CAN**CEL = **<u>*NO</u>** / ***AT**(...)

***AT**(...)
DATE = **<u>*TODAY</u>** / ***TOMOR**ROW / <date 8..10>
,**TIME** = **<u>23:59</u>** / <time 1..8>

,**DATA-ENC**RYPTION = **<u>*NO</u>** / ***YES** / ***ONLY-DATA-INTEGRITY**

,**REC**ORD-**SIZE** = **<u>*NOT-SPECIFIED</u>** /<integer 1..32756>

,**REC**ORD-**FORM**AT = **<u>*STD</u>** / ***FIXED** / ***VARIABLE** / ***UNDEFINED**

(part 5 of 5)

```
,TARGET-FILE-FORMAT = *SAME / *BLOCK-ORIENTED / *SEQUENTIAL(...)
    *SEQUENTIAL(...)
        RECORD-FORMAT = *SAME / *UNDEFINED
,PROTECTION = *STD / *SAME
,LAST-CHANGE-DATE = *STD / *SAME
```

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 84.

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..54> / <c-string 1..512 with-low> / *POSIX(NAME = <posix-pathname 1..510>)
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*

At this point, you must specify the name of the (DVS/POSIX) file in the local system:

– The DVS 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 being executed (see TRANSFER-ADMISSION operand). In this case, SHARE=YES must be set for the file.

  If the file name is specified in the form *$filename*, openFT adds the standard user ID in BS2000 (usually $TSOS) to form *$standarduserid.filename*.

– openFT offers 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.

– The POSIX file name can be specified relative to the HOME directory (entry: ./file) if the file is cataloged in the directory associated with the user ID for which the file transfer is being performed (see also the TRANSFER-ADMISSION operand). If the file for which the file transfer is being executed is not cataloged in the directory of the user ID, the POSIX file name must be entered with the full pathname (entry: /file).

*Specifications **with** pre- or post-processing on FILE-NAME*

– If you specify a pre-processing command when sending, the output from the pre-processing command is sent to the standard output (SYSLST) 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 take the form of a SAM-V file. Since the file attributes are not known at the time the request is accepted, you should note the following if using %TEMPFILE:
  – either the future file attributes must be specified as request attributes,
  – or the file must be transferred in transparent format (homogeneous).

– If you specify a post-processing command when receiving, the received file is used as input for the post-processing command. This file can be addressed using the variable %TEMPFILE. If %TEMPFILE is not specified, then read-in is done via SYSDTA. In this case, the file must be a SAM-V or ISAM-V file.

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 (‘;’). If the pre-/post-processing is running in BS2000, a slash ‘/’ must be placed in front of each individual command. There must not be any blanks between the semicolon and the slash.

Example

```
FILE-NAME=C'|/Command1;/Command2;/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 37. Also refer to the topic “Pre-processing” at the example starting on page 403.

If an error occurs during command execution (in BS2000/JOB-STEP), transfer is aborted with message FTR2083 or FTR2084.

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 37.

Example

```
FILE-NAME = C'|&/command1;/command2;/command3; ...'
```

FILE-NAME = *LIBRARY-ELEMENT(...)
A library member is to be transferred.
The operands in these parentheses can be used as positional operands without their keywords.

LIBRARY =
Name of the library in the local system.

LIBRARY = *NOT-SPECIFIED
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..54>
Name of the library in the local system.

ELEMENT =
Name of the library member in the local system.

ELEMENT = *NOT-SPECIFIED
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>(...)
Name of the library member in the local system.

VERSION =
Version of the element in the local system.

VERSION = *STD
Highest version of the member.

VERSION = <text 1..24>
Version of the member.

TYPE =
Type of the member in the local system.

TYPE = *NOT-SPECIFIED
Only when FTAC functionality is used in the remote system can the library member type be predefined in an FT profile. The type of library member does not have to be specified in the command.

TYPE = <name 1..8>
Type of the member in the local system.

PASSWORD =
Password authorizing access to the file in the local system. If the file in the local system is password-protected , 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 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 = <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
Password authorizing access to the file in the local system.

PASSWORD = *SECRET
The system prompts you to enter the password. However, the input is not displayed on the screen.

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.

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.

TRANSFER-ADMISSION = *SECRET
The system prompts you to enter the transfer admission. However, the input is not displayed on the screen.

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..8>
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 = <c-string 1..8> / <c-string 9..32> / <x-string 1..16>
Password that authorizes the user to access the local system.

PASSWORD = *SECRET
The system prompts you to enter the password. However, the input is not displayed on the screen.

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.

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 BS2000 system.
The default account number of the user ID specified in the USER-IDENTIFICATION is used.

ACCOUNT = <alphanum-name 1..8>
Account number 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 = <c-string 1..8> / <c-string 9..32> / <x-string 1..16>
Password that authorizes the user to access the local system.

PASSWORD = *SECRET
The system prompts you to enter the password. However, the input is not displayed on the screen.

Follow-up processing in the local system

A command sequence can also be input for SUCCESS-PROCESSING and FAILURE-PROCESSING. Each individual command must be preceded by a slash (/). 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.

Follow-up processing operates in BS2000 under the user’s own TSN. Temporary files and temporary job variables can only be accessed for follow-up processing if they belong to this TSN. Access to temporary files of job variables of the TSN under which the actual FT request was carried out is not possible. It is however possible to access permanent files and job variables.

openFT generates a file in BS2000 for the execution of follow-up processing with the name $*userid*.S.FT-BS2.*instance name*.*transfer-id*.JOB, which is automatically deleted after the ENTER-JOB. This file is created by openFT with the catalog attribute `DESTROY=YES` and started with the following operands:

```
ERASE=YES,FLUSH=NO,RERUN=NO
```

If a job class JBCLJOB is set for the user whose ID is being used for the follow-up processing, the follow-up processing is executed under this job class. Otherwise the user’s default job class will be used.

Use of variables in follow-up processing

Variables can be specified within the command or command sequence for the follow-up processing. They are replaced at the start of follow-up processing in the system with values taken from the TRANSFER-FILE command.

The following variables are available:

%PARTNER	for symbolic partner names
%FILENAME	for file name
%ELEMNAME	for element name
%ELEMVERS	for element version
%ELEMTYP	for element type
%RESULT	for the request result
%JOBCLASS	for job class

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.

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.

You can start the follow-up processing under one of the job classes which you have selected by adding the following command in the follow-up processing:

```
/REMARK %JOBCLASS=<jobclass>
```

The following conditions apply for the metastring %JOBCLASS=<jobclass>
- no blanks are permitted within the metastring,
- the metastring must be located immediately before a semicolon (command separator) or at the end of a command sequence,
- the metastring must be after a REMARK or something similar, since it is not deleted before the follow-up processing is performed and otherwise will be mistaken for an invalid follow-up processing command.

openFT checks if the job class is available for the user. If it is, then it is used for the follow-up processing. If it isn't, then the default job class or JBCLJOB is used.

If no direct or indirect (e.g. default) specification is made in the command for a variable which occurs in the follow-up processing data, then it is removed from the follow-up processing data and not replaced.

Logging the follow-up processing

openFT initiates follow-up processing as an ENTER job. The tracer listing is only printed out as standard if an error occurs.
The command LOGOFF with the option SYSTEM-OUTPUT=PRINT or SYSTEM-OUTPUT=DELETE in the follow-up processing allows the user to decide if a tracer listing should be printed out or not.

The follow-up processing job is set up as follows:

```
/.OPENFT LOGON
/SET-FT-INSTANCE <current instance>
/<usercommand>
/<usercommand>
/.........
/SKIP-COMMAND TO-LABEL=FTJOBOK
/SET-JOB-STEP
/LOGOFF
/.FTJOBOK LOGOFF NOSPOOL
```

This set-up only leads to the logging of the follow-up processing job if there are errors in the user commands. If error-free operation is to be logged as well, then the last user command must be LOGOFF.
Note that follow-up processing always runs in the instance in which the TRANSFER-FILE command was executed.

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>
BS2000 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>
BS2000 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.

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:

S.FT-BS2.*instance name*.*transfer-id*.LST

Here, *inst* stands for the instance name of the current openFT instance and t*ransfer-id*. for the identification number of the FT request.
This reference is output to the screen when acceptance of the request is confirmed.

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:

S.FT-BS2.*instance name*.*transfer-id*.LST

For more information see LISTING=*LISTFILE on page 376.

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.

MONJV =
Specifies whether the FT request is to be monitored using a job variable.

MONJV = *NONE
The FT request is not monitored using a job variable.

MONJV = <filename 1..54>
Name of the job variable that is to monitor the transfer.

JV-PASSWORD =
Specifies whether a password is required in order to access the job variable.

JV-PASSWORD = *NONE
No password is required for the job variable.

JV-PASSWORD = <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
Password that is required for the job variable.

JV-PASSWORD = *SECRET
The system prompts you to enter the password. However, the input is not displayed on the screen.

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 predefined by XHCS.

CODED-CHARACTER-SET = <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.
In this case the default values of the operands for the remote system correspond to those specified for the local system. LOGON passwords are not defaulted, however.

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:	*SAME	*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 56 for details).

FILE-NAME = *SAME
relevant for *BS2000.
The file or the library member in the remote system has the same name as the file in the local system.

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.
relevant for *MSP:
The entry FILE-NAME=*NOT-SPECIFIED is only useful if an exit routine is installed in the remote system and the TRANSFER-ADMISSION for the remote system is an FT transfer admission. *NOT-SPECIFIED means in this case that the file name is taken from this exit routine and must not be specified in the command.
Default value if the remote system is an OS/390 or MVS computer.

FILE-NAME = <filename 1..54> / <c-string 1..512 with-low> /
***POSIX(NAME = <posix-pathname 1..510>)**
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.

*Specifications **without** pre- or post-processing on FILE-NAME*

relevant for *BS2000 and *ANY
*POSIX syntax can only be used for BS2000 partners.

For DVS file names in the remote BS2000 system (send or receive file), the following applies:

– With *BS2000 , 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 ($userid.filename) if the file is not cataloged under the user ID for which the file transfer is performed (TRANSFER-ADMISSION operand).
– If an FT product is used in the remote BS2000 and this product carries out the customary BS2000 extension of file names of the form *$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.

For POSIX file names in the remote BS2000 system (send or receive file), the following applies:

– The POSIX file name can be specified relative to the HOME directory (entry: ./file) if the file is cataloged in the directory associated with the user ID for which the file transfer is being performed. If the file for which the file transfer is being executed is not cataloged in the directory of the user ID, the POSIX file name must be entered with the full pathname (entry: /file) (see also the TRANSFER-ADMISSION operand).

*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.

Since the file attributes are not known at the time the request is accepted, you should note the following if using %TEMPFILE:
– either the future file attributes must be specified as request attributes,
– or the file must be transferred in transparent format (homogeneous).

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 (';'). If the pre-/post-processing is running in BS2000, a slash '/' must be placed in front of each individual command. There must not be any blanks between the semicolon and the slash.

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 37.

If an error occurs while executing the commands (in BS2000 /JOB-STEP), 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 37.

Example

```
FILE-NAME = C'|&/Command1;/Command2;/Command3; ...'
```

FILE-NAME = <text 1..56>
relevant for *MSP.
Name of the file in remote OS/390 or z/OS system (send or receive file):

- 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-NAME =<text 1..56>, library members in OS/390 or z/OS can also be addressed, if the FT products used in the partner system support the transfer of library members.

FILE-NAME = *LIBRARY-ELEMENT(...)

REMOTE-PARAMETER =	*BS2000	*MSP	*ANY
only relevant for:	X	[1]	X
default value:	*SAME		*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.
If library members are transferred onto library members, and if these members are administered in both systems with the library management program LMS, then the FT request must contain the same member type for both the local and the remote system.

Furthermore, the remote system must be capable of processing library members.

LIBRARY =
Name of the library in the remote system.
For file transfer with FTAM partners, only library members In the local BS2000 system can be accessed.

LIBRARY = *SAME
relevant for BS2000.
The library name in the remote system is the same as the library name in the local system. This entry is only permitted if the operand LIBRARY-ELEMENT was specified in the local 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 = *SAME
relevant for *BS2000.
The member name in the remote system is identical to the member name in the local system. This entry is only permitted for BS2000 systems, and only if the operand LIBRARY-ELEMENT was specified in the local 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 = *SAME
relevant for *BS2000.

The version in the remote system is identical to the version in the local system. If a file name has been specified in the local system, then the highest version of the member is assumed as the entry for 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 = *SAME
relevant for *BS2000.
The member type in the remote system is identical to the member type in the local system. If a file name is specified in the local system, then TYPE must not be specified.

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:	*SAME	*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 = *SAME
relevant for *BS2000.
The same password applies in the remote system as in the local system.

PASSWORD = *NONE
relevant for *BS2000, *MSP and *ANY.
Access is possible without a password.

PASSWORD = *SECRET
relevant for *BS2000, *MSP and *ANY.
The system prompts you to enter the password. However, this does not appear on the screen.

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:	*SAME	[1]	*NONE

[1] For OS/390 or z/OS systems, there is no default value due to the lack of FTAC functionality. As a rule, the entry TRANSFER-ADMISSION=*PARAMETERS(...) will be necessary.

TRANSFER-ADMISSION = *SAME
relevant for *BS2000.
The relevant values from the local system apply. A password, however, is only accepted if it is explicitly specified in the LOCAL-PARAMETER structure.

TRANSFER-ADMISSION = *NONE
relevant for *ANY.
The remote system does not require/recognize any transfer admission.

TRANSFER-ADMISSION = *SECRET
relevant for *BS2000, *MSP and *ANY.
The system prompts you to enter the transfer admission. However, the input is not displayed on the screen.

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.
If an exit routine exists in the remote OS/390 or z/OS system which offers an FTAC transfer admission, this can be specified using TRANSFER-ADMISSION.

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..7>	<c-string 1..67 with-low>
ACCOUNT=	*NONE <alphanum-name 1..8>	*NONE <text 1..43>	*NONE <c-string 1..64 with-low>

REMOTE-PARAMETER =	***BS2000**	***MSP**	***ANY**
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.

PASSWORD = *SECRET
relevant for *BS2000, *MSP and *ANY.
The system prompts you to enter the password. However, the entry does not appear on the screen.

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.

In file transfer with FTAM partners follow-up processing cannot be started in the remote system.
Exception: an admission profile defines follow-up processing and is addressed via the TRANSFER-ADMISSION). Thus the PROCESSING-ADMISSION operand is not effective for FTAM partners.

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..7>	<c-string 1..67 with-low>
ACCOUNT =	*NONE <alphanum-name 1..8>	*NONE <text 1..43>	*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.

PASSWORD = *SECRET
The system prompts you to enter the password. However, the entry does not appear on the screen.

Follow-up processing in the remote system

A command sequence can also be input for SUCCESS-PROCESSING and FAILURE-PROCESSING.
Each individual command must be preceded by a slash (/).
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

If an ENTER-JOB is initiated in follow-up processing, this job is subject to the usual settings for the user ID which initiated the follow-up processing.

Follow-up processing operates in BS2000 under the user’s own TSN. Temporary files and temporary job variables can only be accessed for follow-up processing if they belong to this TSN. Access to temporary files of job variables of the TSN under which the actual FT request was carried out is not possible. It is however possible to access permanent files and job variables.

openFT generates a file in BS2000 for the execution of follow-up processing with the name $*userid*.S.FT-BS2.*instance name*.*transfer-id*.JOB, which is automatically deleted after the ENTER-JOB. This file is created by openFT with the catalog attribute `DESTROY=YES` and started with the following operands:

```
ERASE=YES,FLUSH=NO,RERUN=NO
```

If a job class JBCLJOB is set for the user whose ID is being used for the follow-up processing, the follow-up processing is executed under this job class. Otherwise the user’s default job class will be used.

As of openFT V8.0 for BS2000, the special form of follow-up processing, *DELETE, is available for requests which move files between hosts. This character string can be specified as remote follow-up processing in the case of receive requests provided that the BS2000 partner system is running openFT V8.0 for BS2000 or, in the case of FTAM partners, openFT V6.0 for Windows or Unix systems is running.

*DELETE causes openFT itself to delete the sent file 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, *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.

Use of variables in follow-up processing

Variables can be specified within the command or command sequence for the follow-up processing. They are replaced at the start of follow-up processing in the system with values taken from the TRANSFER-FILE command.

The following variables are available:

%PARTNER	for symbolic partner names
%FILENAME	for file name
%ELEMNAME	for element name
%ELEMVERS	for element version
%ELEMTYP	for element type
%RESULT	for the request result
%JOBCLASS	for job class

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.

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.

You can start the follow-up processing under one of the job classes which you have selected by adding the following command in the follow-up processing:

```
/REMARK %JOBCLASS=<jobclass>
```

The following conditions apply for the metastring %JOBCLASS=<jobclass>:
– no blanks are permitted within the metastring,
– the metastring must be immediately before a semicolon (command separator) or at the end of a command sequence,
– the metastring must be after a REMARK or something similar, since it is not deleted before the follow-up processing is performed and otherwise will be mistaken for an invalid follow-up processing command.

openFT checks if the job class is available for the user. If it is, then it is used for the follow-up processing. If it isn't, then the default job class or JBCLJOB is used.

If no direct or indirect (e.g. default) specification is made in the command for a variable which occurs in the follow-up processing data, then it is removed from the follow-up processing data and not replaced.

Logging the follow-up processing

openFT initiates follow-up processing as an ENTER job. By default, the tracer listing is only printed out as standard if an error occurs.
The command LOGOFF with the option SYSTEM-OUTPUT=PRINT or SYSTEM-OUTPUT=DELETE in the follow-up processing allows the user to decide if a tracer listing should be printed out or not.

The follow-up processing job is set up as follows:

```
/.OPENFT LOGON
/SET-FT-INSTANCE <current instance>
/<usercommand>
/<usercommand>
/.........
/SKIP-COMMAND TO-LABEL=FTJOBOK
/SET-JOB-STEP
/LOGOFF
/.FTJOBOK LOGOFF NOSPOOL
```

This set-up only leads to the logging of the follow-up processing job if there are errors in the user commands. If error-free operation is to be logged as well, then the last user command must be LOGOFF.
Note that follow-up processing always runs in the instance in which the TRANSFER-FILE command was executed.

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.

Follow-up processing data cannot be transferred to FTAM partners. Exception: SUCC=' *DELETE' for receive requests and openFT-AC is used. If FTAC is used in the remote system, this restriction can be avoided by creating an admission profile in the remote system that defines 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.

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.

No follow-up processing data can be transferred to FTAM partners. If FTAC is used in the remote system, this restriction can be avoided by creating an admission profile in the remote system that defines 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= <name 1..8>
Coding (CCS) that is to be used to read or write the remote file. The character set must be known in the remote system.

FTAM-specific operands

Due to the support of the FTAM protocol, the number of different partner systems is considerably larger when FTAM functionality is used for openFT. For each case, a check should be carried out before the first “productive” file transfer to determine the interoperability between the local system and the partner system (the degree to which they can work together). It is also advisable to perform this check even if the partner system has passed the conformance test.

For an FTAM partner, it is advisable to enter REMOTE-PARAMETER=*ANY(...) in the TRANSFER-FILE command. In this case, openFT will not check the syntax of the entries for the remote system. Of course, these entries must conform to the syntax rules of the remote system. All value assignments in round brackets must be enclosed by quotes.

The operands ACCESS-MODE, FILE-AVAILABILITY, LEGAL-QUALIFICATION and CREATE-PASSWORD are only for communication with FTAM partners. openFT thus supports the operands prescribed in the FTAM norm. With these operands, you can set the attributes of the target file when you make a file transfer request.

These operands are ignored for requests with openFT partners. The file transfer is executed, however.

Functions which cannot be used with FTAM partners

The following functions may not be used for communication with FTAM partners:

– transfer of library members in the remote system.
– transfer of follow-up data into the remote system.
– transfer of ISAM and PAM files.

If you attempt to use such a function in a file transfer request, the request is rejected with the message FTR2170:

```
FTR2170  Request (&00). Remote system: Function not supported(&02)
```

FILE-AVAILABILITY =
Availability of the destination file.
The parameter *availability* can have one of two values: *immediate* or *deferred*. A file may be *deferred* if it has been archived, for example. The partner is responsible for interpreting the term *deferred*. The FTAM partner conventions must therefore be observed here.
Requests involving FTAM partners that do not support the storage group are rejected. In this case, the request is executed, but the entry for FILE-AVAILABILITY is ignored.

FILE-AVAILABILITY = *BY-RECEIVING-SYSTEM
The FILE-AVAILABILITY file attribute is set to a system-specific default value. In BS2000, this is the value IMMEDIATE.

FILE-AVAILABILITY = *IMMEDIATE
The destination file attribute is set to *immediate*.

FILE-AVAILABILITY = *DEFERRED
The destination file attribute is set to *deferred*.

STORAGE-ACCOUNT =
Account number for file storage.
The operand is not available for requests involving FTAM partners that do not support the storage group. In this case, the request is executed, but the entry for STORAGE-ACCOUNT is ignored.

STORAGE-ACCOUNT = *NONE
No account number is specified.

STORAGE-ACCOUNT = <c-string 1..40 with-low> / <text 1..40>
With FTAM partners, this indicates the storage account. File storage fees are to be charged to this account. This operand must be set in accordance with partner system conventions.

ACCESS-MODE =
This sets the access rights of the destination file, provided the security group is available. It is possible to restrict certain access rights in the generated file. The file transfer is only executed if this is permitted by the newly set access rights. If this is not the case, the receive file is created in the remote system, but the transfer is not carried out.
If the access rights cannot be set, or cannot be set in the specified combination, the remote partner may nonetheless be able to carry out the file transfer.

ACCESS-MODE = *BY-RECEIVING-SYSTEM
The default values of the FTAM partner system apply.

ACCESS-MODE = *PARAMETERS(...)
Indicates permitted access methods.

READ-FILE = *NO / *YES
The file cannot or can be read.

INSERT-DATA-UNIT = *NO / *YES
Data units, such as records, cannot or can be inserted in the file.

REPLACE-FILE = *NO / *YES
The file cannot or can be overwritten.

EXTEND-FILE = *NO / *YES
The file cannot or can be extended.

ERASE-DATA-UNIT = *NO / *YES
Data units, such as records, cannot or can be deleted from the record.

READ-ATTRIBUTES = *NO / *YES
The file attributes cannot or can be read.

CHANGE-ATTRIBUTES = *NO / *YES
The file attributes cannot or can be changed.

DELETE-FILE = *NO / *YES
The file cannot or can be deleted.

LEGAL-QUALIFICATION =
Legal qualification.
The operand is not available for requests involving FTAM partners that do not support the security group. In this case, the request is executed, but the LEGAL-QUALIFICATION entry is ignored.

LEGAL-QUALIFICATION = *NONE
There are no legal agreements.

LEGAL-QUALIFICATION = <c-string 1..80 with-low> / <text 1..80>
With FTAM partners, this specifies a legal qualification for the file (similar to a copyright). This must not exceed 80 characters.

CREATE-PASSWORD =
You can use CREATE-PASSWORD to assign a password to send requests for systems which demand password authorization prior to file creation.

CREATE-PASSWORD = *NONE
Creates a file without a password.

CREATE-PASSWORD = <c-string 1..64 with-low> / <x-string 1..128>
Password authorizing a user to create a file in a remote system.
The password must be specified in quotes.

CREATE-PASSWORD = *SECRET
When the system prompts you to enter a password. The input is not echoed on screen.

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).

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 (e.g. with FTAM partners), 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 (e.g. with FTAM partners), 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 (this is only possible with SAM files in BS2000).

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 permitted in BS2000 partners if

– the send file is a SAM file,
– 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.

DATA-TYPE = *USER
The send file contains structured binary data of variable record length. 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).
Only SAM files and PLAM library members can be transferred with DATA-TYPE= *CHARACTER.

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.
Any file that is not a SAM file or a PLAM library member is always transferred as a binary file.

TRANSPARENT =
Specifies if the file is to be converted to a transparent format.
If a file is received in transparent format then openFT for BS2000 ≥ V6.0 automatically sets it up with its original attributes.

TRANSPARENT = *NO
No transparent format should be generated.

TRANSPARENT = *YES
The file should be sent transparently. openFT will reject the transfer of a file in transparent format in the following cases:
– with simultaneous specification of WRITE-MODE=*EXT (FTR2042 or FTR2166)
– if a file in transparent format is to be picked up and the partner system doesn't support this function (FTR2040),
– if the receive file is a library member (FTR2087 or FTR2210).
– if a file is transferred in transparent format to a library member (FTR2216 or FTR2096).

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.
This entry is valid if the user has the appropriate authorization for the entry.
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 Request (&00). File structure error(&02) or
% FTR2210 Request (&00). Remote system: File structure error(&02)
```

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.
Binary files with fixed record lengths (in which the file consists of records of equal lengths) can only be transferred to an FTAM partner if this supports variable length records for binary files.

RECORD-FORMAT = *VARIABLE
The data is transferred in variable length records.
Binary files in user format (in which a record consists of a record length field and the data itself) can only be transferred in the form of variable length records to an FTAM partner if this supports variable length records for binary files.

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 (as SAM-U files in BS2000/OSD systems).

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.

PROTECTION =
Controls the transfer of protection attributes if the partner is a BS2000 system.

PROTECTION =*STD
Only the default file attributes are transferred (behavior up to V10).

PROTECTION =*SAME
The protection attributes USER-ACCESS, ACCESS, BASIC-ACL, EXPIRATION-DATE, FREE-FOR-DELETION and DESTROY are additionally transferred. This requires openFT as of V11 to be used in the partner system.

If the openFT instances of the two partners are using different versions, only those file attributes are transferred that are supported in both versions.

In all cases, the following requirements apply
– the openft protocol is used for transfer
– the source and target file are DMS files
– the target file is not a file generation
– the target file is created or overwritten
– the transfer is not carried out in transparent mode.

LAST-CHANGE-DATE =
Controls whether the last change date of the send file (i.e. date+time) is accepted for the receive file.

LAST-CHANGE-DATE = *STD
The current time is accepted as the change date of the receive file. This corresponds to the behavior up to openFT V11.0.

LAST-CHANGE-DATE =*SAME
The change date of the send file is accepted as the change date of the receive file. This function is supported only for the openFT protocol. In BS2000/OSD it is also necessary to use OSD V8.0 or higher.

If the target system does not support acceptance of the change date or if no modification date is sent be the source system, no request with LAST-CHANGE-DATE=*SAME is executed and an error is reported.

If acceptance of the change date matches the default behavior of the target system, the parameter is ignored.

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 (CANCEL-FILE-TRANSFER) command), or you can get information on the status of the FT request (SHOW-FILE-TRANSFER 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 260).

If the file transfer was successful, openFT outputs an the following asynchronous message as a result message (if the user process is still active and allows asynchronous messages):

```
% FTR0005 Request (&00). File '(&02)' transferred
```

Command return codes

For a list of the possible return codes, see the tables as of page 433.

4.32.3 Examples of the TRANSFER-FILE command

This section provides sample applications of the TRANSFER-FILE command.

1. TRANSFER-FILE command for openFT with mandatory operands only

   When the conditions on page 362 apply, the TRANSFER-FILE 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.

```
/TFF TO,HOST001,(DATA)
%  FTR0000 Request (&00) accepted
/
```

   The long form of this command is as follows:

```
/TRANSFER-FILE TRANSFER-DIRECTION=TO,                                    -
/              PARTNER=HOST001,                                  -
/              LOCAL-PARAMETER=(FILE-NAME=DATA)
```

   If the user operating under this user ID continues to operate in interactive mode, he/she receives the following asynchronous message after successful file transfer:

```
%MESS %  FTR0005 (&01'')Request (&00). File '(&02)' transferred
```

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 PASS1492.

   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 *LISTE.002*, which has not yet been created. If this file already exists, the LIST file should not be transferred.

   Recommended short form of the command:

```
/TRANS-FILE FROM,HOST002,                                                -
/           (LIST.002),                                                  -
/           (LIST,'XX',TRANS-AD=(SHIPPING,SHIP002,'PASS1492')),WRITE=NEW
%  FTR0000 Request (&00) accepted
```

Long form of the command:

```
/TRANSFER-FILE TRANSFER-DIRECTION=FROM-PARTNER,PARTNER=HOST002,     -
/              LOCAL-PARAMETER=(FILE-NAME=LIST.002),                 -
/              REMOTE-PARAMETER=*BS2000(FILE-NAME=LIST,PASSWORD=C'XX',   -
/              TRANSFER-ADMISSION=(USER-IDENTIFICATION=SHIPPING,         -
/              ACCOUNT=SHIP002,PASSWORD='PASS1492')),WRITE-MODE=*NEW-FILE
%  FTR0000 Request (&00) 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* and printed out there. For retrieval of the data, the transfer admission GETREPORT has been set up on each branch computer.

   The transfer of these files is carried out with the following procedure:

```
/BEGIN-PROCEDURE LOGGING=NO,PARAMETERS=YES(PROCEDURE-PARAMETERS=       -
/(&MONTH=),ESCAPE-CHARACTER='&')
/REMARK PLEASE SPECIFY THE PREVIOUS MONTH FOR &MONTH !
/TRANS-FILE FROM,BRANCH1,                                              -
/(REPORT.&MONTH..BRANCH1,                                              -
/SUCC='/PRINT-FILE FILE-NAME=REPORT.&MONTH..BRANCH1,                    -
/SPOOLOUT-NAME=BRANCH1,LAYOUT-CONTROL=(CONTROL-CHARACTERS=EBCDIC)'),    -
/(REPORT.&MONTH,TRANS-AD=GETREPORT)
/TRANS-FILE FROM,BRANCH2,                                              -
/(REPORT.&MONTH..BRANCH2,                                              -
/SUCC='/PRINT-FILE FILE-NAME=REPORT.&MONTH..BRANCH2,                    -
/SPOOLOUT-NAME=BRANCH2,LAYOUT-CONTROL=(CONTROL-CHARACTERS=EBCDIC)'),    -
/(REPORT.&MONTH,TRANS-AD=GETREPORT)
/TRANS-FILE FROM,BRANCH3,                                              -
/(REPORT.&MONTH..BRANCH3,                                              -
/SUCC='/PRINT-FILE FILE-NAME=REPORT.&MONTH..BRANCH3,                    -
/SPOOLOUT-NAME=BRANCH3,LAYOUT-CONTROL=(CONTROL-CHARACTERS=EBCDIC)'),    -
/(REPORT.&MONTH,TRANS-AD=GETREPORT)
/TRANS-FILE FROM,BRANCH4,                                              -
/(REPORT.&MONTH..BRANCH4,                                              -
/SUCC='/PRINT-FILE FILE-NAME=REPORT.&MONTH..BRANCH4,                    -
/SPOOLOUT-NAME=BRANCH4,LAYOUT-CONTROL=(CONTROL-CHARACTERS=EBCDIC)'),    -
/(REPORT.&MONTH,TRANS-AD=GETREPORT)
/TRANS-FILE FROM,BRANCH5,                                              -
/(REPORT.&MONTH..BRANCH5,                                              -
/SUCC='/PRINT-FILE FILE-NAME=REPORT.&MONTH..BRANCH5,                    -
/SPOOLOUT-NAME=BRANCH5,LAYOUT-CONTROL=(CONTROL-CHARACTERS=EBCDIC)'),    -
/(REPORT.&MONTH,TRANS-AD=GETREPORT)
/END-PROCEDURE
```

This call procedure (name: DO.MONTH) is called as follows:

```
/CALL-PROCEDURE DO.MONTH
%/REMARK PLEASE SPECIFY THE PREVIOUS MONTH FOR &MONTH !
%&MONTH=november
%  FTR0000 Request 194577 accepted
%  FTR0000 Request 194987 accepted
%  FTR0000 Request 195442 accepted
%  FTR0000 Request 196099 accepted
%  FTR0000 Request 196583 accepted
/
```

4. Distribution of files

   A central office distributes guidelines to its five branch offices.
   It has the guidelines printed out as soon as they are received in the branch. All user IDs concerned in the central office and in the branch offices are CENTRAL, have the account number CENTR01 and the password C'CEN'.

   To distribute the guidelines the central office uses the following ENTER-JOB procedure:

```
/LOGON
/TRANS-FILE TO,BRANCH1,                                                    -
/(GUIDELINE,TRANS-AD=(CENTRAL,CENTR01,C'CEN')),                            -
/(SUCC='/PRINT-FILE FILE-NAME=GUIDELINE,SPOOLOUT-NAME=IMPORTNT,            -
/LAYOUT-CONTROL=(CONTROL-CHARACTER=EBCDIC)')
/TRANS-FILE TO,BRANCH2,                                                    -
/(GUIDELINE,TRANS-AD=(CENTRAL,CENTR01,C'CEN')),                            -
/(SUCC='/PRINT-FILE FILE-NAME=GUIDELINE,SPOOLOUT-NAME=IMPORTNT,            -
/LAYOUT-CONTROL=(CONTROL-CHARACTER=EBCDIC)')
/TRANS-FILE TO,BRANCH3,                                                    -
/(GUIDELINE,TRANS-AD=(CENTRAL,CENTR01,C'CEN')),                            -
/(SUCC='/PRINT-FILE FILE-NAME=GUIDELINE,SPOOLOUT-NAME=IMPORTNT,            -
/LAYOUT-CONTROL=(CONTROL-CHARACTER=EBCDIC)')
/TRANS-FILE TO,BRANCH4,                                                    -
/(FILE=GUIDELINE,TRANS-AD=(CENTRAL,CENTR01,C'CEN')),                       -
/(SUCC='/PRINT-FILE FILE-NAME=GUIDELINE,SPOOLOUT-NAME=IMPORTNT,            -
/LAYOUT-CONTROL=(CONTROL-CHARACTER=EBCDIC)')
/TRANS-FILE TRANS=TO,PARTNER=BRANCH5,                                      -
/(GUIDELINE,TRANS-AD=(CENTRAL,CENTR01,C'CEN')),                            -
/(SUCC='/PRINT-FILE FILE-NAME=GUIDELINE,SPOOLOUT-NAME=IMPORTNT,            -
/LAYOUT-CONTROL=(CONTROL-CHARACTER=EBCDIC)')
/LOGOFF
```

   This ENTER-JOB procedure (name: ENTER.GUIDELINE) is called as follows:

```
/.CENTRAL ENTER-JOB ENTER.GUIDELINE,TIME=99
%  JMS0066 JOB ACCEPTED. TSN=2083
```

The BS2000 command SHOW-USER-STATUS can be used to trace the sequence of this ENTER process and the file transfer operations:

```
/SHOW-USER-STATUS
NAME      TSN TYPE   PRI    CPU-USED CPU-MAX ACCOUNT#
CENTRAL  2083 1      210      0.0         99 CENTR01
GP46518  2065 3 DIA  210      0.4595   32767 CENTR01
/SHOW-USER-STATUS
NAME      TSN TYPE   PRI    CPU-USED CPU-MAX ACCOUNT#
OPENFT 2088 1      210      0.0          1 CENTR01
GP46518  2065 3 DIA  210      0.6868   32767 CENTR01
NAME      TSN TYPE   PRI    SIZE  COPIES CPU-USED
CENTRAL  2083 4 PR   210       2       0
/SHOW-USER-STATUS
NAME      TSN TYPE   PRI    CPU-USED CPU-MAX ACCOUNT#
GP46518  2065 3 DIA  210      0.7957   32767 CENTR01
OPENFT 2099 2 BAT  210      0.0938       1 CENTR01
OPENFT 2100 2 BAT  210      0.0548       1 CENTR01
NAME      TSN TYPE   PRI    SIZE  COPIES CPU-USED
CENTRAL  2083 4 PR   210       2       0
OPENFT 2091 4 PR   210       1       0
OPENFT 2096 4 PR   210       1       0
OPENFT 2097 4 PR   210       1       0
/SHOW-USER-STATUS
NAME      TSN TYPE   PRI    CPU-USED CPU-MAX ACCOUNT#
GP46518  2065 3 DIA  210      0.8545   32767 CENTR01
NAME      TSN TYPE   PRI    SIZE  COPIES CPU-USED
CENTRAL  2083 4 PR   210       2       0
OPENFT 2091 4 PR   210       1       0
OPENFT 2096 4 PR   210       1       0
OPENFT 2097 4 PR   210       1       0
OPENFT 2101 4 PR   210       1       0
OPENFT 2102 4 PR   210       1       0
```

5. Job transfer and job processing

   To reduce the load on the central office host computer an ASSEMBLER program is to be compiled in the host computer of a branch office. To do this, the program is compiled within a CALL-PROCEDURE procedure file that is started as a follow-up processing operation. After processing of the FT request has been completed, the compiled program is stored in a module library. The module library is then to be transferred back to the central host computer.

   The CALL-PROCEDURE procedure DO.JOB could look as follows:

```
/BEGIN-PROCEDURE LOGGING=COMMANDS
/DELETE-FILE FILE-NAME=*
/ASSIGN-SYSDTA TO-FILE=*SYSCMD
/START-ASSEMBH
   .
   .   Program to be compiled!
   .
/START-LMS
//OPEN-LIB CENTRAL.MODLIB,MODE=*UPDATE
//ADD-ELEMENT FROM-FILE=*OMF,TO-ELEMENT=*LIB-ELEM(TYPE=R)
//END
/ASSIGN-SYSDTA TO-FILE=*PRIMARY
/TRANS-FILE TRANS=TO,PARTNER=CENTRAL,LOC=(FILE-NAME=CENTRAL.MODLIB,      -
/           TRANS-AD=(CENTRAL,CENTRO1,C'ZEN'))
/END-PROCEDURE
```

   This CALL-PROCEDURE procedure is transferred to the computer in branch office 1 using the following command and started there. The request is to be monitored in the central host by job variable XYZ.

```
/TRANS-FILE TO,BRANCH1,(DO.JOB,MONJV=XYZ)REM=,                           -
/        (TRANS-AD=(CENTRAL,CENTRO1,C'CEN'),SUCC='/CALL-PROCEDURE DO.JOB')
%  FTR0000 Request 194572 accepted
```

6. 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 files to be collected are password-protected. The user IDs are not password-protected (The operating system BS2000 has been generated with system parameter SSMCOPT=Y.

   The files are transferred using the following CALL-PROCEDURE procedure (name: *DO.KETTE*):

```
/BEGIN-PROCEDURE LOGGING=COMMANDS
/TRANS-FILE TRANS=FROM,PARTNER=BRANCH1,WRITE=EXT,                        -
/REM=(FILE-NAME=SAM.FILE,PASS='FIL1',TRANS=GETFILES),                    -
/LOC=(FILE-NAME=SAM.ALL,                                                 -
/SUCC='/TRANS-FILE TRANS=FROM,PARTNER=BRANCH2,WRITE=EXT,                 -
/REM=(FILE-NAME=SAM.FILE,PASS=''FIL2'',TRANS=GETFILES),                  -
/LOC=(FILE-NAME=SAM.ALL,                                                 -
/SUCC=''/TRANS-FILE TRANS=FROM,PARTNER=BRANCH3,WRITE=EXT,                -
/LOC=(FILE-NAME=SAM.ALL),                                                -
/REM=(FILE-NAME=SAM.FILE,PASS=''''FIL3'''',TRANS=GETFILES)'')')
/END-PROCEDURE
```

   The procedure is called as follows:

```
/CALL-PROCEDURE DO.KETTE
%/BEGIN-PROCEDURE LOGGING=COMMANDS
%/TRANS-FILE TRANS=FROM,PARTNER=BRANCH1,WRITE=EXT,
REM=(FILE-NAME=SAM.FILE,PASS='FIL1',TRANS=GETFILES),
LOC=(FILE-NAME=SAM.ALL,SUCC='/TRANS-FILE TRANS=FROM,
PARTNER=BRANCH2,WRITE=EXT,REM=(FILE-NAME=SAM.FILE,
PASS=''FIL2'',TRANS=GETFILES),
LOC=(FILE-NAME=SAM.ALL),
SUCC=''/TRANS-FILE TRANS=FROM,PARTNER=BRANCH3,WRITE=EXT,
LOC=(FILE-NAME=SAM.ALL),
REM=(FILE-NAME=SAM.FILE,PASS=''''FIL3'''',TRANS=GETFILES)'')')
%  FTR0000 Request 164572 accepted
%/END-PROCEDURE
```

7. File transfer between openFT for BS2000 and another FT system

   The file FILE is to be transferred for BS2000 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:

```
/TRANS-FILE TO,SYS,(FILE),                                          -
/*ANY('f/i/l/e',TRANS-AD=('BROOKLYN','20000'))
%  FTR0000 Request 165572 accepted
```

8. Transfer of POSIX files between two openFT for BS2000 systems

   The POSIX file named ‘file’ is to be transferred to the home directory /home/JIMMY in the BS2000 computer HOST12 using openFT. The user ID JIMMY with the account number 12345678 and password 2000 are used in HOST12. As the user ID for the remote system is specified, it would have been sufficient to specify the relative path name. The short form of the command is used:

```
/TRANS-FILE TO,HOST12,(‘,/file‘), -
/(c‘/home/JIMMY/file‘,TRANS-ADM=(JIMMY,12345678,'20000'))
%  FTR0000 Request 165581 accepted
```

   Command with *POSIX:

```
/TRANS-FILE TO,HOST12,(*POSIX(./file)), --
/(*POS(/home/JIMMY/file),TRANS-ADM=(JIMMY,12345678,'20000'))
%  FTR0000 Request 165581 accepted
```

9. File transfer to z/OS systems

   The file FILE is to be transferred from a BS2000 computer to the OS/390 or z/OS computer IBM1. The file is held in the local system under the name CENTRBS2 and has the password C’ZZZZ’. The user ID has the account number CENTR01 and is protected by the password C’CEN1’. The file is to be called FILE in the OS/390 or z/OS computer, have the password FILE01, and be stored under the user ID CENMSP with the account number CENTRAL02 and password CEN2. The command is entered in the short form.

```
/TRANS-FILE TO,IBM1,                                                -
/(FILE,C'ZZZZ',TRANS-AD=(CENTRBS2,CENTR01,C'CEN1')),                -
/*MSP(FILE,FILE01,TRANS-AD=(CENMSP,CENTRAL02,CEN2))
%  FTR0000 Request 143581 accepted
```

10. File transfer using openFT for Unix systems

    The file mailbag is to be obtained from the Unix system ALFRED and transferred to the local BS2000 system. The file is held by ALFRED under the user ID „flyte“ with a password of 144002 and is to be given the name NOHURRY in the local system. An abbreviated command for this request is as follows:

```
/TRANS-FILE FROM,ALFRED,(NOHURRY),*ANY('mailbag',                    -
/TRANS-AD=('flyte',,'144002'))
%  FTR0000 Request 122181 accepted
```

11. File transfer with subsequent follow-up processing, but with no report on the execution of the follow-up processing

    The file COBOL.LIST is to be transferred from the computer with the name COMPILER to the local system and printed out there four times. This file is stored in the COMPILER computer under the user ID COBOL with account number COBACC and the password PASSWORD. The report on the execution of the print process is to be suppressed if the operation is successful. There are two ways of carrying out these operations. In the first, more detailed method a procedure FT.PRINT.PROC is initially set up in the local system containing the following:

```
/BEGIN-PROCEDURE LOGGING=NO,PARAMETERS=YES(PROCEDURE-PARAMETERS=      -
/(&FILE=,&.COPIES=0)
/PRINT-FILE FILE-NAME=&FILE,REPEAT=&COPIES,                          -
/LAYOUT-CONTROL=(CONTROL-CHARACTERS=EBCDIC)
/END-PROCEDURE
```

    The actual file transfer request has the format

```
/TRANS-FILE FROM,COMPILER,                                           -
/           (COBOL.LIST,                                             -
/           SUCC='/CALL-PROC FT.PRINT.PROC,                          -
/           (FILE=COBOL.LIST,COPIES=3)'),                            -
/           (TRANS-AD=(COBOL,COBACC,'PASSWORD'))
```

    However, the CALL-PROCEDURE procedure is not required if a command is specified in the TRANS-FILE command for the follow-up processing. The file transfer request then has the following format:

```
/TRANS-FILE FROM,COMPILER,                                           -
/           (COBOL.LIST,SUCC='/PRINT-FILE FILE-NAME=%FILENAME        -
/           REPEAT=3,LAYOUT-CONTROL=(CONTROL-CHARACTERS=EBCDIC)'),   -
/           (TRANS-AD=(COBOL,COBACC,'PASSWORD'))
```

12. Procedure file transfer with subsequent execution of the procedure as follow-up processing and finally deletion of the transferred file

    The procedure file PROC.JOB, which is constructed in exactly the same way as the file DO.JOB in example 5, is to be transferred to the computer BRANCH1 and executed there under the user ID CENTRALL with the account number CENTRALCENTR01 and the password CEN. Then the file is to be deleted in the remote system.

    The requisite command is as follows:

```
/TRANSFER-FILE                                                    -
/             TRANSFER-DIRECTION=TO,                              -
/             PARTNER=BRANCH1,                                    -
/             LOCAL-PARAMETER=(FILE-NAME=PROC.JOB),               -
/             REMOTE-PARAMETER=(TRANSFER-ADMISSION=(CENTRAL,CENTR01, -
/             'CEN'),SUCCESS-PROCESSING='/CALL-PROCEDURE %FILENAME;  -
/             /DELETE-FILE %FILENAME')
```

    A possible short form of this command would be:

```
/TRANS-FILE TO,BRANCH1,(PROC.JOB),(TRANS-AD=(CENTRAL,CENTR01,'CEN'),-
/SUCC='/CALL-PROC %FILENAME;/DEL-FILE %FILENAME')
```

13. File transfer using FTAC

    The file TURNOVER is to be transferred to the computer JACKJOHN. On this computer openFT for BS2000 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 ‘Fordearsteven’ 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.
    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:

```
/TRANSFER-FILE                                                    -
/             TRANSFER-DIRECTION=TO,                              -
/             PARTNER=JACKJOHN,                                   -
/             LOCAL-PARAMETER=(FILE-NAME=TURNOVER),               -
/             REMOTE-PARAMETER=*BS2000(FILE-NAME=*NOT-SPECIFIED,  -
/             TRANSFER-ADMISSION='Fordearsteven')
```

There is of course a short form:

```
/TFF TO,JACKJOHN,(TURNOVER),(*NOT-SPEC,                -
/           TRANS-AD='Fordearsteven')
```

14. Time-driven file transfer

    A user from the West End,
    was really at his wits' end.
    He wanted his FT
    to be at ten fifty
    and so this command he did send:

```
/TRANSFER-FILE TO,WESTEND2,(FILE),START=(,10:50)
/    TRANS-ADMISSION='GreetingsFromWestEnd'
```

    With this command the file FILE is transferred to computer WESTEND2 and stored there under the same name with the same ID as in the local system. The start time for this request is set at the earliest for 10.50 on the day the request is issued. This is why there is a comma first in the parentheses after START. It indicates that for the DATE operand that should come first in the parentheses, the default value *TODAY has been taken.

    Since the ID is password-protected, it is necessary to specify the transfer admission GreetingsFromWestEnd. If the IDs are not password-protected then this operand is omitted from the command.

15. Local file processing between two openFT for BS2000 systems

    A list of the names of files for the local ID is to be transferred to the remote file SFA-FILE.LOCAL.

```
/TRANSFER-FILE                                                  -
/   TRANSFER-DIRECTION=*TO-PARTNER, PARTNER=BS2PART,            -
/   LOCAL-PARAMETER=(FILE-NAME=                                 -
/   C'|/SHOW-FILE-ATTRIBUTES OUT=*SYSLST'),                     -
/   REMOTE-PARAMETER=*BS2000(FILE-NAME=SFA-FILE.LOCAL,          -
/   TRANSFER-ADMISSION=PROFBS2PART)
%  FTR0000 Request 172281 accepted
```

Short form:

```
/TFF TO,BS2PART,('|/SHOW-FI-ATTR OUT=*SYSLST'),                -
/   (SFA-FILE.LOCAL,,PROFBS2PART)
```

16. Remote pre-processing between two openFT for BS2000 systems

    A list of the FT partner systems in the remote system is to be transferred to the local file INFO.BS2PART.

    In openFT versions <10 the remote ID must possess the FT-ADM privilege in order to be able to obtain information about the partner systems. If this is not the case then the request is aborted with message FTR2140.

```
/TRANSFER-FILE                                                 -
/   TRANSFER-DIRECTION=*FROM-PARTNER, PARTNER=BS2PART,         -
/   LOCAL-PARAMETER=(FILE-NAME=INFO.BS2PART),                  -
/   REMOTE-PARAMETER=*BS2000(FILE-NAME=                        -
/   C‘|/SH-FILE-ATTR OUT=*SYSLST;/SH-FT-PART OUT=*SYSLST‘,     -
/   TRANSFER-ADMISSION=PROFBS2PART)
%  FTR0000 Request 132281 accepted
```

    Short form:

```
/TFF FROM,BS2PART,(INFO.BS2PART),                              -
/ ('|/SH-FILE-ATTR OUT=*SYSLST;/SH-FT-PART OUT=*SYSLST'        -
/ ,,PROFBS2PART)
```

17. 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=*BS2000(FILE-NAME=                        -
/   C‘|ftinfo -csv‘,                                           -
/   TRANSFER-ADMISSION=PROFUNKNOWN)
%  FTR0000 Request 132274 accepted
```

    Short form:

```
/TFF FR,UNKNOWN,(FTINFO.UNKNOWN),('|ftinfo -csv',,PROFUNKNOWN)
```

18. Local pre-processing and remote post-processing

    Information on the last 100 locally written logging records should be stored in the remote BS2000 system PARTBS2, in the file FILE.LOG under the access admission TRANSADM. The FILE.LOG file should be write-protected after the transfer.

```
/TFF TO,PARTBS2,('|/SH-FT-LOG ,100,OUT=*SYSLST'),              -
/     ('|/CPF %TEMPFILE,FILE.LOG;/MDFA FILE.LOG,PROT=(,*READ)' -
/     ,,TRANSADM)
```

4.33 TRANSFER-FILE-SYNCHRONOUS Transfer file synchronously

Note on usage

User group: FT user

Alias name: FTSCOPY

Functional description

With the TRANSFER-FILE-SYNCHRONOUS 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 TRANSFER-FILE command. Consequently only the syntax is described.

TRANSFER-FILE-SYNCHRONOUS differs from TRANSFER-FILE 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 local parameters LISTING, MONJV and JV-PASSWORD are not supported.

– The general parameters PRIORITY, START and CANCEL are not used, because they do not have any significance for synchronous transfer.

The name of the command procedure which previously had the identical name has changed to START-TRANSFER-FILE-SYNCH.

Format

(part 1 of 4)

TRANSFER-**FILE-SYN**CHRONOUS / **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..54> / <c-string 1..512 with-low> /
***LIB**RARY-**ELEM**ENT(...) / ***POS**IX(NAME=<posix-pathname 1..510>)

***LIB**RARY-**ELEM**ENT(...)

LIBRARY = ***NOT-SPECIFIED** / <filename 1..54>

,**ELEM**ENT = ***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>

,**PASS**WORD = ***NONE** / <c-string 1..4> / <x-string 1..8> /
<integer -2147483648..2147483647> / ***SECRET**

,**TRANS**FER-**ADM**ISSION = ***SAME** / <alphanum-name 8..32> / <x-string 15..64> /
<c-string 8..32 with-low> / ***SECRET** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

***PAR**AMETERS(...)

USER-IDENTIFICATION = <name 1..8>

,**ACCOUNT** = ***NONE** / <alphanum-name 1..8>

,**PASS**WORD = ***NONE** / <c-string 1..8> / <c-string 9..32> / <x-string 1..16> / ***SECRET**

,**CODED-CHAR**ACTER-**SET** = ***STD** / <name 1..8>

(part 2 of 4)

```
,REMOTE-PARAMETER = *BS2000(...) / *MSP(...) / *ANY(...)
    *BS2000(...)
        FILE-NAME = *SAME / *NOT-SPECIFIED / <filename 1..54> / <c-string 1..512 with-low> /
                    *LIBRARY-ELEMENT(...) / *POSIX(NAME=<posix-pathname 1..510>)
            *LIBRARY-ELEMENT(...)
                LIBRARY = *SAME / *NOT-SPECIFIED / <filename 1..54>
                ,ELEMENT = *SAME / *NOT-SPECIFIED /  <filename 1..64 without-gen-vers>(...) /
                           <composed-name 1..64 with-under>(...)
                    <filename>(...) / <composed-name>(...)
                        VERSION = *SAME / *STD / <text 1..24>
                ,TYPE = *SAME / *NOT-SPECIFIED / <name 1..8>
        ,PASSWORD = *SAME / *NONE / <c-string 1..4> / <x-string 1..8> /
                    <integer -2147483648..2147483647> / *SECRET
        ,TRANSFER-ADMISSION = *SAME / <alphanum-name 8..32> / <x-string 15..64> /
                              <c-string 8..32 with-low>/ *SECRET / *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> / *SECRET
        ,PROCESSING-ADMISSION = *SAME / *NOT-SPECIFIED / *PARAMETERS(...)
            *PARAMETERS(...)
                USER-IDENTIFICATION = <c-string 1..8>
                ,ACCOUNT = *NONE / <alphanum-name 1..8>
                ,PASSWORD = *NONE / <c-string 1..8> / <c-string 9..32> / <x-string 1..16> / *SECRET
        ,SUCCESS-PROCESSING = *NONE / <c-string 1..1000 with-low>
        ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
        ,FILE-AVAILABILITY = *BY-RECEIVING-SYSTEM / *IMMEDIATE / *DEFERRED
        ,ACCESS-MODE = *BY-RECEIVING-SYSTEM / *PARAMETERS(...)
            *PARAMETERS(...)
                READ-FILE = *NO / *YES
                ,INSERT-DATA-UNIT = *NO / *YES
                ,REPLACE-FILE = *NO / *YES
                ,EXTEND-FILE = *NO / *YES
                ,ERASE-DATA-UNIT = *NO / *YES
                ,READ-ATTRIBUTES = *NO / *YES
                ,CHANGE-ATTRIBUTES = *NO / *YES
                ,DELETE-FILE = *NO / *YES
        ,CODED-CHARACTER-SET = *STD / <name 1..8>
```

(part 3 of 4)

```
*MSP(...)
    FILE-NAME = *NOT-SPECIFIED / <text 1..56>
    ,PASSWORD = *NONE / <alphanum-name 1..8> / *SECRET
    ,TRANSFER-ADMISSION = <alphanum-name 8..32> / <x-string 15..64> / <c-string 8..32 with-low> /
                          *SECRET / *PARAMETERS(...)
        *PARAMETERS(...)
            USER-IDENTIFICATION = <name 1..7>
            ,ACCOUNT = *NONE / <text 1..43>
            ,PASSWORD = *NONE / <alphanum-name 1..8> / *SECRET
    ,PROCESSING-ADMISSION = *SAME / *NOT-SPECIFIED / *PARAMETERS(...)
        *PARAMETERS(...)
            USER-IDENTIFICATION = <name 1..7>
            ,ACCOUNT = *NONE / <text 1..43>
            ,PASSWORD = *NONE / <alphanum-name 1..8> / *SECRET
    ,SUCCESS-PROCESSING = *NONE /<c-string 1..1000 with-low>
    ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
    ,CODED-CHARACTER-SET = *STD / <name 1..8>
*ANY(...)
    FILE-NAME = *NOT-SPECIFIED / <filename 1..54> / <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> / *SECRET
    ,TRANSFER-ADMISSION = *NONE / <alphanum-name 8..32> / <x-string 15..64> /
                          <c-string 8..32 with-low> / *SECRET / *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> / *SECRET
    ,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> / *SECRET
```

(part 4 of 4)

```
        ,SUCCESS-PROCESSING = *NONE /<c-string 1..1000 with-low>
        ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
        ,FILE-AVAILABILITY = *BY-RECEIVING-SYSTEM / *IMMEDIATE / *DEFERRED
        ,STORAGE-ACCOUNT = *NONE / <c-string 1..40 with-low> / <text 1..40>
        ,ACCESS-MODE = *BY-RECEIVING-SYSTEM / *PARAMETERS(...)
            *PARAMETERS(...)
                READ-FILE = *NO / *YES
                ,INSERT-DATA-UNIT = *NO / *YES
                ,REPLACE-FILE = *NO / *YES
                ,EXTEND-FILE = *NO / *YES
                ,ERASE-DATA-UNIT = *NO / *YES
                ,READ-ATTRIBUTES = *NO / *YES
                ,CHANGE-ATTRIBUTES = *NO / *YES
                ,DELETE-FILE = *NO / *YES
        ,LEGAL-QUALIFICATION = *NONE / <c-string 1..80 with-low> / <text 1..80>
        ,CREATE-PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128> / *SECRET
        ,CODED-CHARACTER-SET = *STD / <name 1..8>
,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
,DATA-ENCRYPTION = *NO / *YES / *ONLY-DATA-INTEGRITY
,RECORD-SIZE = *NOT-SPECIFIED /<integer 1..32756>
,RECORD-FORMAT = *STD / *FIXED / *VARIABLE / *UNDEFINED
,TARGET-FILE-FORMAT = *SAME / *BLOCK-ORIENTED / *SEQUENTIAL(...)
    *SEQUENTIAL(…)
        RECORD-FORMAT = *SAME / *UNDEFINED
,PROTECTION = *STD / *SAME
```

Operands

The meaning of the operands is the same as for asynchronous file transfer, see the operand description for TRANSFER-FILE as of page 368.

Example

The posix file *file.posix* 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.

```
 TRANS-FILE-SYNC TRANS-DIR=TO,PARTNER=PUX,      -
*LOCAL=*PAR(FILE-NAME=*POS(file.posix)),            -
*REM=*ANY(FILE-NAME='dir/file.ux',TRANS='ForUXSys'), -
COMP=*ZIP
FTR0005 Request 91339. File './file.posix' transferred
```

Short form:

```
FTSCOPY TO,PUX,('./file.posix'),*a('dir/file.ux',,'ForUXSys'),*ZIP
```

5 What to do if...

The error FTR2046 oder FTR2047 is displayed as a direct response to TRANSFER-FILE (NCOPY)

Possible cause	Suggested action
Incorrect transfer admission in the local system	Enter a valid transfer admission
If openFT-AC is installed:	
Transfer admission in the local system does not grant access to the desired request	Use /SHOW-FT-LOGGING-RECORDS␣(RECORD-TYPE=(FT=NONE)),NUMBER=n to output the return code (RC) of the FTAC logging record, and then check the reason for the rejection using /HELP-MSG␣MSG-ID= FTCxxxx, LANG=E (xxxx=RC)

The error FTR0014 occurs in other situations

Possible cause	Suggested action
Incorrect transfer admission in the remote system	Enter a valid transfer admission
Transfer admission in the remote system does not grant access to the required request	Determine the grounds for rejection in the remote system; With BS2000 partners: /SHOW-FT-LOGGING-RECORDS␣ (RECORD-TYPE=(FT=NONE)),NUMBER=n and /HELP-MSG␣MSG-ID= FTCxxxx, LANG=E (xxxx=RC from logging record) With Unix partners: ftshwl␣-rt=c and fthelp␣xxxx (xxxx=RC from logging record) With Windows partners: The logging window and status line show the RC.

The error FTR2212 is displayed following the file management command

Possible cause	Suggested action
Problem in the remote system: remote system is not active	Contact system administrator of remote system
FT not started in the remote system	
FT limits in the remote system have been reached, e.g. CONN-LIM in a remote BS2000 system	
Network problems	Inform network administrator

The error FTR2109 (openFT protocol) resp. FTR0108 (FTAM) is displayed following the file management command

Possible cause	Suggested action
No BCIN or BCACT was entered for the remote system	Inform the system administrator

The error FTR2025, FTR2076 oder FTR2199 is displayed following the file management command to an FTAM partner

Possible causes	Suggested action
The local or remote file management system reports an error which cannot be mapped onto any significant FTR message. The return code output is reported by the protocol, not by the file management system in question.	Try to use a normal COPY command to copy the (send) file to the specified receive file. Any error message which occurs in the process can be related to the FT command.

The error FTRxxxx is displayed as a direct response to TRANSFER-FILE (NCOPY)

Possible cause	Suggested action
Error in the local system	Use /HELP-MSG␣MSG-ID= FTRxxxx, LANG=E to request additional information or consult the manual.

The error FTRxxxx in other situations

Possible cause	Suggested action
Error in the remote system (in most cases) Error in the local system (less common) Exception with the error: FTR0035 / FTR 2077 / FTR 2200 FILE LOCKED both the local and the remote file may be locked. FTR2043 and FTAM partner: rights for permitted actions may be missing in local system.	Use /HELP-MSG␣MSG-ID= FTRxxxx, LANG=E to request additional information. Use /MODIFY-FILE-FT-ATTRIBUTES to change rights

The follow-up processing should always be printed

Possible cause	Suggested action
	Define follow-up processing as an ENTER procedure using /LOGOFF or use the corresponding command sequence

Follow-up processing is not executed

Possible cause	Suggested action
– Error in follow-up processing command and follow-up processing log was intercepted and suppressed . – A space character after a semicolon, if several follow-up processing sessions were specified.	Check follow-up processing commands or prevent error interception in order to receive a log indicating the cause of the problem.

Pre- or post-processing is not being performed

Possible cause	Suggested action
Error in the pre- or post-processing command or a space character after a semicolon, if several pre-/post-processing commands were specified.	Check the pre-/post-processing commands.
The job class, in which the processing job is to be started does not allow processing to be immediately carried out	
The SPOOL sub-system has not started	Start the subsystem SPOOL (have it started)

Remote follow-up processing in event of error (FAILURE PROCESSING) is not executed

Possible cause	Suggested action
TRANSFER-FILE was accepted (FTR0000), but an error was discovered before file transfer started (e.g. receive file locked)	

No result information is displayed at the terminal

Possible cause	Suggested action
No input after the TRANSFER-FILE (NCOPY) command	Enter data at the system
Message overlooked, or a LOGOFF/ LOGON command was entered after the TRANSFER-FILE (NCOPY) command	View logging records
Asynchronous messages have been suppressed with MODIFY-MSG-OPTIONS␣ OPERATOR-MSG=NO	Specify /MODIFY-MSG-OPTIONS␣OPERATOR-MSG=YES
A program has been loaded as a background task	Terminate the program

A cancelled request is still in the request queue

Possible cause	Suggested action
The request was already active when the connection to the remote system was lost	Wait until the connection is restored so that openFT can notify the partner system about the cancellation. If the request does not “disappear” after restoring the connection, it can ultimately be deleted by the FT administrator using CNFT with FORCE-CANCELLATION=*YES.

The FT subsystem or FTAC hang

Possible cause	Suggested action
The subsystem cannot be unloaded because some tasks are still attached.	MOD-SUBSYSTEM-PARAMETERS FORCED=ALLOWED and STOP-SUBSYSTEM FORCE=YES will unload the subsystem, regardless of any running tasks. This may subsequently result in system dumps, which should be ignored. In the case of FTAC, this may be due to the fact that an FTP server is “attached” to FTAC. If this is the case, then it can be terminated with STOP-FTP-DEMON.
The subsystem is in a LOCKED state because the holder task generated a system dump, for example.	UNLOCK-SUBSYSTEM (DSSM version 3.6 and later) unloads the subsystem. Only use this in an absolute emergency, since the request files of all openFT instances, and possibly the user files as well, remain locked. The locks can be removed by the system administrator using VERIFY.

Despite issuing the CANCEL-FILE-TRANSFER (NCANCEL) command, the request has been executed

Possible cause	Suggested action
CANCEL-FILE-TRANSFER (NCANCEL) came too late; the request had already been executed	

A very large file cannot be transferred

Possible cause	Suggested action
There is not enough contiguous disk space on a public volume set to effect the transfer	Discuss the problem with the system administrator

No information on a TRANSFER-FILE (NCOPY) request

Possible cause	Suggested action
	Check using /SHOW-FT-LOGGING-RECORDS␣NUMBER=n

No result list

Possible cause	Suggested action
Specify LIST=*NONE (default value)	
Printout lost	
The default public volume set of the request initiator was not attached at the end of transfer	

The file is locked, though the file transfer was completed

Possible cause	Suggested action
	Contact the system administrator; the file lock can be reset by the administrator with the VERIFY command

The request is not displayed with the SHOW-FILE-TRANSFER (NSTATUS) command, although MONJV is set to W or R

Possible cause	Suggested action
JV was changed illegally	Check in the dialog or the result list for the warning FTR0802
The public volume set for MONJV was not available during the update	

The request marked with * remains in WAIT status

Possible cause	Suggested action
Partner deactivated by the FT administrator	If necessary, inform the system administrator

The request marked with ! remains in WAIT status

Possible cause	Suggested action
Connection setup failed:	
remote system not active	Contact system administrator of remote system
FT not active in the remote system	Contact system administrator of remote system
In a BS2000 system there was no BCIN/BCACT for the local system	Contact system administrator of remote system
Maximum number of connections permitted in remote system currently reached.	

The request marked with? remains in WAIT status

Possible cause	Suggested action
There is no BCIN/BCACT for the remote system (in the local system)	If necessary, inform the system administrator
The local system is not entered in the remote FT system	Contact system administrator of remote system
No valid partner key is stored in the local system (STATE RAUTH in SHOW-FT-PARTNERS)	The FT administrator must store a current, public partner system key in the SYSKEY library on the configuration user ID of the local instance (element type D, name: symbolic name of the partner system as in the network descriptions file.
No valid local system key is stored in the remote system (STATE LAUTH in SHOW-FT-PARTNERS)	The local FT administrator must transmit a current, public key to the FT administrator of the partner system, which then must be stored in the appropriate location.

The request not marked remains in WAIT status

Possible cause	Suggested action
Normal waiting time for system resources	wait

5.1 Frequently asked questions

What is the shortest form of the NCOPY command?

In order to send a file to a partner system, it is usually sufficient to issue the following command. The entries to be replaced begin with lowercase letters:

/TFF TO,partner,(file),(,,transAd)

TFF is an abbreviation for the TRANSFER-FILE (NCOPY) command. The same effect is, of course, also achieved with the alias NCOPY, for example.

Any FTAC transfer admission defined in the remote system (e.g. TRANSADM) may be entered for transAd. Alternative input: (user,acc,passwd).

You can also use the same entry for partners on Unix systems, provided, of course, that you do not object to entering the file name in uppercase letters in Unix systems.

The entry will also work for PC partners, provided the file name is syntactically allowed there.

And what if a library member is to be transferred?

/TFF TO,partner,((lib,elem,type)),(,,transAd)

The file name must be replaced by (lib,elem,type). Note, however, that this input form does not apply to Unix and Windows partners, since no libraries are present there. You should therefore enter

/TFF TO,partner,((lib,elem,type)),(file,,transAd)

or

/TFF TO,partner,((lib,elem,type)),A('file',,'transAd')

in the latter cases.

Please note that you should generally always specify only guaranteed abbreviations in procedures (e.g. *ANY instead of A) to remain independent of the current FT version being used.

Do I need to specify LIST=*NONE in NCOPY?

No. This entry is the default setting to suppress the result list.

How do I determine which FT requests have succeeded and which ones have failed?

The logging records output by:

/SHOW-FT-LOG

shows you the result of the last transfer.

If you want to view the last *n* entries, specify:

/SHOW-FT-LOG ,n

The most recent entry is displayed first.

You can also select logging records using different criteria (e.g. partner, file name, etc.). Note that when openFT-AC is used, two entries are recorded for each NCOPY request: the first entry is the FTAC entry, which is identified by a C in the first column, and the second entry, which follows the first, contains the result of the transfer (identified with T).

If you want to see only the results of the transfer, enter:

/SHOW-FT-LOG (REC-TYPE=(,N)),n

The messages FTR2025, FTR2076 oder FTR2199 DMS ERROR return a non-DMS RC as the return code. Why?

These messages are issued whenever the local or remote file management system issues a return code (on a file access error) that cannot be mapped to one of the more informative FTR messages (e.g. FILE UNKNOWN, FILE NOT SHAREABLE). This may be potentially caused by two problems:

The remote file management system need not be the BS2000 DMS (it may be Unix system or a Windows system, for example).

The transmission protocol only provides for standardized return codes, so your file transfer does not receive the original return code generated on the partner - even if the remote system happens to be a BS2000 system.

Consequently: DMS error simply means an error from the (respective!) file management system, and the return code contains the code forwarded by the transmission protocol.

In such cases, it is often worth trying to copy the file with a normal COPY command (possibly on the local and remote system), since the internal system RC would then be received in the event of an error.

How does one detect whether an error has occurred on the local or remote system?

The following rules apply:

If the NCOPY command is not accepted with FTR0000, but is rejected immediately instead, the error always lies on the local system.

For NCOPYs that are rejected after being accepted with FTR0000, the error is almost always on the remote system. As of openFT V10, it is also possible to identify the origin of the problem from the message. If the reason for the rejection is `FTR2169 Remote system: Transfer admission invalid`, the cause in this case always lies in the remote system.

In cases where the partner cannot be reached at all (e.g. FTR0108), the situation is more ambiguous, and there is generally no way of knowing on which side the problem occurred.

How can I easily determine whether or not a partner can be reached?

It is generally not advisable to test an FT connection using the NCOPY command, since the request is processed asynchronously, and the result is therefore not immediately visible. A much simpler test can be performed using:

/SHOW-REM-FILE partner

If the partner cannot be reached, you will immediately receive a corresponding message.

If the partner can be reached, your request will be rejected by the partner with FTR2169 (since you did not specify a transfer admission or specified an invalid transfer admission), with FTR0020 or FTR2027 (since no file was specified) or with FTR2170 (since the partner does not support file management).

This test can be performed independently of the operating system.

Can I determine the name of a file on the remote system?

Yes. The command

/SHOW-REM-FILE partner,*DIR('.'),,transAdm

shows you all files on the partner system, or more precisely, all files that you may access under the specified transmission permission transAdm.

Restriction: The '.' entry is not supported by older FT-BS2000 versions. Use *DIR($userid.) in such cases.

If desired, you can also have all members of a library displayed with:

/SHOW-REM-FILE partner,*DIR('lib/typ'),,transAdm

How can I wait for the result of a transfer before proceeding with a procedure?

By specifying a MONJV, assuming, of course, that your system has monitoring job variables. Enter the command:

/NCOPY TO,partner,(file,MONJV=jv),(,,transAd)

/WAIT (jv,2,1) EQ 'T' OR (jv,2,1) EQ 'A',TIME=sec

NCOPY starts the transmission. The WAIT command then waits for a maximum of sec seconds for the transmission to complete. If the operation terminates normally, the job variable is assigned a 'T' at the second position; if an abort occurs, an 'A'.

Another possibility is to use synchronous transfer with FTSCOPY (TRANSFER-FILE-SYNCHRONOUS).

Why was my FT request rejected even though I entered a correct transfer admission?

It is indeed possible for a request to be rejected despite a correctly specified transfer admission (in the form (user,accout,password), for example) or TRANSADM. This is because your request could also be rejected if the transfer admission does not allow you to execute all the actions you want. Here are some potential reasons:

The user ID is locked on the remote system (e.g. by SEVER/LOCK-USER in BS2000).

The remote system is not allowing any requests which use transfer admissions of the form (user, account,password), since all levels in the FTAC admission set have been set to 0.

The desired direction of transfer or your system was rejected by the partner.

The partner system does not allow the desired function, e.g. follow-up processing or even file management.

In addition, the transfer admission is often specified in uppercase instead of lowercase, and vice versa, especially when given over the phone. Uppercase letters can be effectively specified only within quotes..

Finally, it is possible that the transfer admission you specified was really invalid.

My call was rejected with FTR2169 Remote system: Transfer admission invalid. How do I find out the reason?

The rejection comes from the partner system. Consequently, the cause can only be determined there.

With openFT products, the reason can be easily determined from the FTAC logging record.

To do this, ask your partner to display the last logging record or the last *n* logging records under the receiving ID:

– in BS2000, with: /SHOW-FT-LOG [,n]

– in a Unix system and a Windows system with ftshwl [-nb=n] or via the respective graphical user interface.

Using the partner, file name, time, etc., as reference points, you will first need to look for the matching FTAC entry (type C or FTAC). The reason for the rejection will be given in the RC column. The meaning of the RC is output directly on a PC; it can be explicitly requested with /HELP FTCnnnn in BS2000 and with fthelp nnnn in Unix system or Windows system (where nnnn is the RC).

If your partner cannot find any logging record for your request, you have either not contacted the correct partner, or the specified transfer admission does not belong to the expected receiving ID. This could be primarily because the transfer admission does not exist (especially if you entered it incorrectly, for example).

What is an FT or FTAC transfer admission and how can I set one up?

The normal way to identify oneself on a remote system is via the logon entries, i.e., the user ID, account number and password. TRANSADM=(user-id,account,password).

A simpler method is to use a special authorization exclusively for the file transfer (TRANS-ADM=transAdm). In order to avoid exposing all the details of his/her entire logon authorization, the owner of the transfer admission sets up a so-called admission profile as follows:

– In BS2000: /CREATE-FT-PROFILE name,,transAdm

– In Unix system or a Windows system with ftcrep name transAdm or via the respective graphical user interface with File / New / Admission Profile.

In the above entries, name is the name under which the profile can be administered (e.g. deleted again) and may be up to 8 characters in length. transAdm is the admission which is assigned by the partner and which you specify in your FT command, and must be at least 8 characters. If blanks or other special characters appear in it or if a distinction between uppercase and lowercase letters is to be made, the entry must be enclosed within single quotes.

6 Appendix

6.1 Command return codes for file transfer and file management

The command return codes listed in the table below may occur in connection with file transfer and file management. They apply to the following commands

CREATE-REMOTE-DIR
DELETE-REMOTE-DIR
DELETE-REMOTE-FILE
MODIFY-FILE-ATTRIBUTES
MODIFY-REMOTE-DIR-ATTR
MODIFY-REMOTE-FILE-ATTRIBUTES
TRANSFER-FILE
TRANSFER-FILE-SYNCHRONOUS

(SC2)	SC1	Maincode	Meaning
0	0	FTR0000	Request accepted. The request ID is present as an insert in the SYSOUT message.
0	0	CMD0001	The file transfer request has been successfully completed. Follow-up processing has been started for both the local system and the remote system, as requested, provided no error occurred. Local errors are indicated a message.
20	64	FTR0020	The command was not executed. Send file unknown.
35	128	FTR0035	File locked to prevent multiple access. The send file or the receive file is already locked by another process to prevent it from being updated simultaneously.
41	128	FTR0041	Request queue full.
108	128	FTR0108	Remote system not accessible.
32	32	CMD0221	Request rejected. Internal error. Job variable not accessible.
33	32	CMD0221	Request rejected. Internal error.
34	32	FTR0802	Warning: Job variable contents inconsistent

(SC2)	SC1	Maincode	Meaning
35	64	FTR0803	Request rejected. Follow-up processing could not be started. Incorrect specifications for the admission to perform follow-up processing (processing admission).
36	32	CMD0221	Request rejected. Request data inconsistent.
82	32	CMD0221	Internal error. Job variable not accessible
83	32	CMD0221	Internal error.
84	32	CMD0221	Internal error. Current instance incompatible.
85	32	CMD0221	Internal error. The command could not be executed because *open*FT code could not be loaded.
86	32	CMD0221	Writing of log records no more possible. Process terminated
87	32	CMD0221	No space left on device for internal files
88	32	CMD0221	Error during ops generation
89	32	CMD0221	Error in key file '>>1<<'
90	32	CMD0221	Internal error. Set / release file-locks not possible
94	32	CMD0221	Protocol stack '>>1<<' not installed
95	32	CMD0221	FTAC subsystem not available
231	32	CMD0221	openFT panic >>1<<. Abnormal termination
2	0	FTR2014	No file attribute changes requested
15	64	FTR2015	*open*FT is not authorized to execute requests for this user
16	64	FTR2016	The specified directory is not empty
17	64	FTR2017	File attributes do not match request parameters
18	64	FTR2018	Attributes could not be modified
19	64	FTR2019	File could not be created
21	64	FTR2021	CCS name unknown
22	64	FTR2022	Higher-level directory not found
23	64	FTR2023	The command was not executed. Receive file already exists.
24	64	FTR2024	Transfer of file generation groups not supported
25	64	FTR2025	Error accessing. DMS error, possibly the transfer ID
26	64	FTR2026	Resulting file name too long. The relative file name was specified in the transfer request. The absolute file name completed by *open*FT is longer than permitted.
27	64	FTR2027	No file name specified. Request ID present as insert in the SYSOUT message.
28	64	FTR2028	Invalid management password
29	128	FTR2029	The volume for either the send file or the receive file is not mounted

(SC2)	SC1	Maincode	Meaning
30	64	FTR2030	Home directory not found
31	64	FTR2031	Renaming not possible
128	32	FTR2032	The command was not (fully) executed. Not enough space
33	64	FTR2033	The command was not executed. The owner of the file is unknown in the system
34	64	FTR2034	Invalid file password
36	64	FTR2036	Retention period of file not yet expired
37	64	FTR2037	File is read only
38	64	FTR2038	File structure not supported
39	64	FTR2039	Syntax error in resulting file name
42	64	FTR2042	Extension of file not possible for transparent transfer
43	64	FTR2043	Access to file denied
44	64	FTR2044	Follow-up processing exceeds length limit
45	64	FTR2045	Processing admission invalid
46	64	FTR2046	Local transfer admission invalid
47	64	FTR2047	Request rejected by local FTAC
48	64	FTR2048	The function is not available for the selected protocol
49	64	FTR2049	Remote follow-up processing not supported
50	64	FTR2050	Data integrity check not supported
51	64	FTR2051	User data encryption not possible for this request
70	64	FTR2070	*open*FT is no longer authorized to execute requests for this user
71	64	FTR2071	User data encryption not installed
72	64	FTR2072	The FT request was canceled because, for example, the command CANCEL-FILE-TRANSFER was specified, or the time specified in TRANSFER-FILE has been reached.
73	64	FTR2073	Encryption error
74	64	FTR2074	Request rejected. The file could not be created.
75	64	FTR2075	Higher-level directory no longer found
76	64	FTR2076	Request rejected. I/O error. Receive file may have been deleted.
77	128	FTR2077	File now locked to prevent multiple access 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.
78	128	FTR2078	Request rejected. the volume for either the send file or the receive file is not available

(SC2)	SC1	Maincode	Meaning
79	64	FTR2079	File no longer found. The local send or receive file can no longer be accessed because, for example, it was deleted during an interruption of the *open*FT system.
80	64	FTR2080	Home directory no longer found
81	128	FTR2081	The command was not executed (any further) executed. Not enough space.
82	64	FTR2082	File owner no longer known
83	64	FTR2083	Request rejected. Pre-/post-processing error.
84	64	FTR2084	Request rejected. An exit code for pre-/post-processing is returned.
85	64	FTR2085	Request rejected. File password no longer valid
86	64	FTR2086	File is now read only
87	64	FTR2087	Request rejected. File structure error. Request ID is present as an insert in the SYSOUT message.
88	128	FTR2088	NDMS error The request was rejected because the partner system currently does not have the resources available to accept requests.
89	64	FTR2089	The restart attempts were unsuccessful, for example, a pre-/post-processing command could not be completed before the termination of *open*FT.
90	64	FTR2090	Error in file transfer completion. 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.
2	0	FTR2091	Requests only partially completed; 'n' of 'm' files were transferred. In the case of a synchronous send request with wildcards, not all files were successfully transferred.
92	64	FTR2092	Access to 'file' no longer permissible 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.
93	128	FTR2093	FTAM error 'error_code'.
94	64	FTR2094	Retention period of file not yet expired.
95	64	FTR2095	Extension of file not possible for transparent transfer
96	64	FTR2096	File structure not supported

(SC2)	SC1	Maincode	Meaning
109	64	FTR2109	Connection setup rejected by local transport system
110	64	FTR2110	Data integrity check indicates an error
111	64	FTR2111	Encryption/data integrity check not possible. Encryption switched off.
112	64	FTR2112	Data integrity check not supported by partner
113	64	FTR2113	Request rejected. User data encryption not possible for this request
114	64	FTR2114	Identification of local system rejected by remote system
115	64	FTR2115	Interrupted by remote system
116	64	FTR2116	The local application is not defined in the transport system
117	64	FTR2117	Local application not available
118	64	FTR2118	Authentication of local system failed
119	64	FTR2119	Local system unknown in remote system
120	64	FTR2120	Remote system unknown
121	64	FTR2121	Authentication of partner failed
122	64	FTR2122	FT session rejected or disconnected. Reason ’error’
123	64	FTR2123	OSS call error
124	128	FTR2124	No free connection
125	128	FTR2125	Request rejected. Connection lost
126	64	FTR2126	Transport system error. Error code ’error_code’
127	128	FTR2127	No data traffic within ’n’ seconds
140	64	FTR2140	Remote system: *open*FT is not authorized to execute requests for this user
141	64	FTR2141	Remote system: Directory is not empty
142	64	FTR2142	Remote system: File attributes do not match the request parameters
143	64	FTR2143	Remote system: Attributes could not be modified
144	64	FTR2144	Remote system: File/directory could not be created
145	64	FTR2145	Remote system: CCS name unknown or not supported
146	64	FTR2146	Remote system: Higher-level directory not found
147	64	FTR2147	Remote system: File/directory already exists
148	64	FTR2148	Remote system: Transfer of file generation groups not supported
149	64	FTR2149	Remote system: Access error DMS error, possibly the transfer ID. The FT system continues to run after output of the message

(SC2)	SC1	Maincode	Meaning
150	64	FTR2150	Request rejected. Remote system: Resulting file name too long
151	128	FTR2151	Remote system: File locked to prevent multiple access
152	64	FTR2152	Remote system: No file or directory name specified
153	64	FTR2153	Remote system: Invalid management password
154	128	FTR2154	Remote system: File/directory not available
155	64	FTR2155	Remote system: File/directory not found. On deletion, the return code SC2=1 and SC1=0 is output, otherwise SC2=155 and SC1=64.
156	64	FTR2156	Remote system: Home directory not found
157	64	FTR2157	Remote system: Renaming not possible
158	128	FTR2158	The command was not executed (any further). Remote system: Not enough space
159	64	FTR2159	Remote system: File owner unknown
160	64	FTR2160	Remote system: Invalid file password
161	64	FTR2161	Remote system: Retention period of file not yet expired
162	64	FTR2162	Remote system: File/directory is read only
163	64	FTR2163	Remote system: File structure not supported
164	64	FTR2164	Request rejected. Remote system: Syntax error in resulting file name
165	64	FTR2165	Remote system: Transparent file transfer not supported
166	64	FTR2166	Remote system: Extension of file not possible for transparent transfer
167	64	FTR2167	Remote system: Access to denied
168	64	FTR2168	Remote system: Follow-up processing exceeds length limit
169	64	FTR2169	Request rejected. Remote system: Transfer admission invalid. Transfer admission incorrect or insufficient FTAC authorizations.
170	64	FTR2170	Request rejected. Remote system: Function not supported
171	64	FTR2171	Remote system: Processing admission invalid
172	128	FTR2172	Remote system: Request queue full
195	64	FTR2195	Remote system: *open*FT is not longer authorized to execute requests for this user
196	64	FTR2196	The request has been canceled in the remote system
197	64	FTR2197	Request rejected. Remote system: File/directory could not be created
198	64	FTR2198	Remote system: Higher-level directory no longer found

(SC2)	SC1	Maincode	Meaning
199	64	FTR2199	Remote system: I/O error, e.g. DMS error, possibly the transfer ID. The file was deleted during transfer.
200	128	FTR2200	Remote system: File now locked to prevent multiple access
201	128	FTR2201	Request rejected. Remote system: File/directory no longer available
202	64	FTR2202	Remote system: File/directory no longer found
203	64	FTR2203	Remote system: Home directory no longer found
204	128	FTR2204	The command was not executed (any further). Remote system: File/directory gets no more space
205	64	FTR2205	Remote system: File owner no longer known
206	64	FTR2206	Request rejected. Remote system: Pre-/post-processing error
207	64	FTR2207	Request rejected. Remote system: Exit code is returned during pre-/post-processing
208	64	FTR2208	Request rejected. Remote system: File password no longer valid
209	64	FTR2209	Remote system: File/directory is now read only
210	64	FTR2210	Request rejected. Remote system: File structure error.
211	128	FTR2211	Remote system: NDMS error
212	64	FTR2212	Recovery failed
213	128	FTR2213	Remote system: Resource bottleneck
214	64	FTR2214	Remote system: Access to ’file’ is no longer permissible
215	128	FTR2215	FTAM error
216	64	FTR2216	Remote system: File structure not supported
217	64	FTR2217	Remote system: Retention period of file not yet ex
218	64	FTR2218	Remote system: Extension of file not possible for transparent transfer
2	0	FTR2225	Information output canceled
226	64	FTR2226	Job variable contents inconsistent
227	64	FTR2227	Job variable not in use by *open*FT
228	64	FTR2228	Job variable not found

SC1/2 = Subcodes 1/2 in decimal notation
See the section entitled Messages for additional information.

6.2 Structure of CSV outputs

6.2.1 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 SHOW-FTAC-ENVIRONMENT. 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.

6.2.2 SHOW-FILE-TRANSFER

The following table indicates the CSV output format of a request.

Short output is also possible with SHOW-FILE-TRANSFER, see page 445.

The **Parameter** column contains the name of the output parameter during long output, see page 272.

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 / *SECRET / empty	LOC: SUCC-PROC
LocFailProc	String	Local follow-up processing on error, enclosed in double quotes / *NONE / *SECRET / empty	LOC: FAIL-PROC
LocListing	String	*SYSLST / *LISTFILE / *NONE Result list in the local system	LOC: LIST

Column	Type	Values and Meaning	Parameter
LocMonjv	String	Name of the job variable enclosed in double quotes / *NONE	LOC: MONJV
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 / *SECRET / *NONE / empty	REM: SUCC-PROC
RemFailProc	String	Remote follow-up processing on error, enclosed in double quotes / *SECRET / *NONE / empty	REM: FAIL-PROC

Column	Type	Values and Meaning	Parameter
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
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	*IMMEDIATE / *DEFERRED / *NSPEC Availability (for FTAM only)	AVAILABILITY
StorageAccount	String	Account number (for FTAM only) / empty	STOR-ACCOUNT
AccessRights	String	FTAM access rights / empty Possible values are @r, @w or combinations of r, i, p, x, e, a, c, d	ACCESS-RIGHTS
LegalQualif	String	Legal qualification (for FTAM only) / empty	LEGAL-QUAL
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

Column	Type	Values and Meaning	Parameter
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 SHOW-FILE-TRANSFER in CSV format

INF=*SUMMARY outputs a table with two rows indicating the number of requests that have the corresponding status, see also page 265.

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

6.2.3 SHOW-FT-ADMISSION-SET

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 279.

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

6.2.4 SHOW-FT-LOGGING-RECORDS

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 450.

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 296ff.

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)

6.2.5 SHOW-FT-MONITOR-VALUES

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 309.

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 / FTAM / 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

```
/SHOW-FT-MONITOR-VALUES NAME=*ALL,OUTPUT=*SYSOUT(*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
```

6.2.6 SHOW-FT-OPTIONS

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 323ff. 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	empty	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 / FTAM / 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 In the case of FTAM, "nil-Application Entity Title" is sent	---
StatName	String	$FJAM	---
SysName	String	Name of the local system (host 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	*STD / port number Port number of the local FTAM server	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 / FTAM / 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	*ACTIVE / *INACT / *DISABLED / *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 / FTAM / 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)

6.2.7 SHOW-FT-PARTNERS

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 333.

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	empty	---
Entity	String	empty	---
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

6.2.8 SHOW-FT-PROFILE

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 342f.

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)	

6.2.9 SHOW-FT-RANGE

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 346.

Column	Type	Values and Meaning	Parameter
SecLev	Number	Security level	SECLEV
PartnerName	String	Partner name	PARTNER-NAME

6.2.10 SHOW-REMOTE-FILE-ATTRIBUTES SHOW-FILE-FT-ATTRIBUTES

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 or *NAME-ONLY (SHOW-FILE-ATTRIBUTES), only the FileName column is output.

The **Parameter** column indicates the name of the output parameter in the case of detailed output, see page 352ff.

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	

Column	Type	Values and Meaning	Parameter	Std
FileAvail	String	*IMMEDIATE / *DEFERRED / *NSPEC File avaliability	FILE-AVAILABILITY	
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	

```
/SHOW-REM-FILE-ATTR BS2MCH01,FILE1,,TRANSADM,INF=*ALL,OUT=*SYSOUT(*CSV)

FileName;StorageAccount;CreIdentity;CreTime;ModIdentity;ModTime;ReaIdentity;
ReaTime;AtmIdentity;AtmTime;FileType;CharSet;RecFormat;RecSize;FileAvail;
AccessRights;FileSize;MaxFileSize;LegalQualif;CcsName
"FILE1";*NSPEC;"MISTERX";*NSPEC;*NSPEC;2012-03-19 12:39:47;*NSPEC;*NSPEC;
*NSPEC;*NSPEC;*NSPEC;*NSPEC;*VAR;*NSPEC;*NSPEC;r-pxeacd---;2048;*NSPEC;
*NSPEC;*NSPEC

/SHOW-REM-FILE-ATTR BS2MCH02,FILE2,,TRANSADM,INF=*STD,OUT=*SYSOUT(*CSV)

FileName;StorageAccount;CreIdentity;ModTime;FileType;AccessRights;FileSize
"FILE2";*NSPEC;"MISTERX";2012-03-19 12:39:47;*NSPEC;r-pxeacd---;2048
```

Glossary

Italic type indicates a reference to other terms in this glossary.

access control
: *File attribute* in the *virtual filestore*, attribute of the *security group* that defines *access rights*.

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.

action list
: Component of the file attribute *access control* (attribute of the *security group*) in the *virtual filestore* that defines *access rights*.

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.

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.

Application Entity Title (AET)

The Application Entity Title consists of Layer 7 addressing information of the *OSI Reference Model*. It is only significant for *FTAM partners*.

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*).

audit

Fundamental function of a secure system; logging of operating sequences and preparation of the logged data.

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 repertoire

Character set of a file in the *virtual filestore*.
In the case of files transferred with *FTAM partners* it is possible to choose between: *GeneralString*, *GraphicString*, *IA5String* and *VisibleString*.

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*

concurrency control

Component of the FTAM file attribute *access control* (part of the *security group*) in the *virtual filestore* that controls concurrent access. openFT for BS2000 offers only passive and partial support for concurrency control. Note: “partial support” is a technical term taken from the FTAM environment that means that the parameter is interpreted correctly at the syntactic level but is not genuinely supported.

configuration user ID

Each openFT instance in BS2000 requires an ID, on which the variable files of this file are stored (for the default instance: $SYSFJAM).

connectivity

In general, the ability of systems and partners to communicate with one another. Sometimes refers simply to the communication possibilities between transport systems.

constraint set

Component of the *document type*.

contents type

File attribute in the *virtual filestore,* attribute of the *kernel group* that describes the file structure and the form of the file contents.

cross domain connection

Connection mode in which a TRANSDATA network is connected as an SNA domain to an SNA domain via a *gateway*.

DASD (Direct Access Storage Device)

Disk storage

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.

DHCP

Service in TCP/IP networks that automatically assigns IP addresses and TCP/IP parameters to clients on request.

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. In BS2000 (DVS), PLAM libraries are interpreted as directories.

document type

Value of the file attribute *contents type* (attribute of the *kernel group*). Describes the type of file contents in the *virtual filestore*.

– *document type* for text files: FTAM-1
– *document type* for binary files: FTAM-3

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.

entity

see *instance*

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 directory / file catalog

File present in every *pubset* (in SM pubsets there is a file directory in every volume set). All a pubset's files and job variables are entered in the corresponding *file directory*. Files on private disks and tapes can be entered in the file directory.
A catalog entry contains all a file's or job variable's attributes (protection attributes, location of the administered data etc.

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.

filestore, virtual

see *virtual filestore*

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

Administrator of the FTAC functions; should be identical to the person responsible for data security in the system.

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 file attributes

All systems which permit file transfer via FTAM protocols must make their files available to their partners using a standardized description (ISO 8571). To this end, the attributes of a file are mapped from the physical filestore to a *virtual filestore* and vice versa. This process distinguishes between three groups of file attributes:

– kernel group: describes the most important file attributes.
– storage group: contains the file's storage attributes.
– security group: defines security attributes for file and system access control.

FTAM partner

Partner system that uses *FTAM protocols* for communication.

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.

functionality class

Class which places certain minimum security function demands on an IT system.
The functionality classes are defined in the “Kriterien für die Bewertung der Sicherheit von Systemen der Informationstechnik (IT)”, (Criteria for the assessment of the security of Information Technology (IT) systems), version 1 of 11th January 1989, published by the Zentralstelle für Sicherheit in der Informationstechnik (Central Office for Security in Information technology) on behalf of the German government.

functional standard

Recommendation defining the conditions and the forms of application for specific ISO standards (equivalent term: *profile*). The transfer of unstructured files is defined in the European Prestandard CEN/CENELEC ENV 41 204; file management is defined in the European Prestandard CEN/CENELEC ENV 41205.

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.

general string

Character repertoire for file files transferred to and from *FTAM partners*.

global privileges

All the privileges that can be assigned using the /SET-PRIVILEGE command including the security administrator privilege and the TSOS privilege. Global privileges and *system administrator privileges* are identical.

global request identification / global request ID / global request number

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.

global user administration

This comprises the administration of user IDs and user groups and covers resources and user rights, the creation, modification and deletion of user IDs and user groups

GraphicString

Character repertoire for files transferred to and from *FTAM partners*.

guard

A component of the GUARDS condition administration system. A guard unites conditions which are evaluated by the standard GUARDS condition administration system on request.

GUARDS (Generally Usable Access Control Administration System)

Object administration for *Guards*.

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.

host

Formerly a large-scale data processing system which required a *front-end processor* in order to be able to communicate. Nowadays, the term used for BS2000 or z/OS systems.

IA5String

Character repertoire for files transferred to and from *FTAM partners*.

identification
Procedure making it possible to identify a person or object.

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.

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).

job
Sequence of commands, statements and data.

job class

Job classes combine *jobs* which share certain properties and characteristics.

job transfer

Transfer of a file that constitutes a *job* in the *receive system* and is initiated as a job there.

joinfile / user catalog / user ID catalog

File that contains the *user attributes* of all the *user IDs* in a *pubset*.

kernel group

Group of file attributes of the *virtual filestore* that encompasses the kernel attributes of a file.

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.

maximum-string-length

Specifies the maximum length of *strings* within a file in the *virtual FTAM filestore*.

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.

NEA

Name of a network architecture.

network description file

File used up to openFT V9 that contains specifications concerning *remote systems* (*FT systems*).

Network Management Kernel

Component of the Network Management Platform; responsible for forwarding network management requests as well as for centralized tasks such as logging, authorization checks, request and application administration.

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 the HIPLEX cluster. Each instance has its own address (instance ID, virtual BCAM 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 is always the ID under which the request is submitted, not the ID under which it is executed.

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.

permitted actions

File attribute in the *virtual filestore*; attribute of the *kernel group* that defines actions that are permitted in principle.

Personal Audit for Individual Accountability

Trace of individual system utilization. Identification can take the following forms:

– a user ID corresponds to a user, or
– a user may use only one operator terminal.

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.

presentation

Entity that implements the presentation layer (layer 6) of the *ISO/OSI Reference Model* in an *FT system* that uses e.g. *FTAM protocols*.

presentation selector

Subaddress used to address a *presentation application*.

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*.

privilege

– Global privilege within the system that authorizes a user to execute certain commands and call certain program interfaces (e.g. TSOS privilege).
– Set of user-specific attributes that are used by the access control system.

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*.

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.

public space

Named disk storage area which is available to a defined number of user IDs within the operating system. This storage area may be located on one or more Public Volume Sets (*pubsets*).

pubset / public volume set

Set of shared, named disk storage units which is defined by a catalog identification (catid). A distinction is made between *SF pubsets* and SM pubsets.

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

OSI term for an element in a layer that acts as an intermediary between two other partners and thus makes communications between these two partners possible.
In the narrow sense, on the network layer a relay is the functional equivalent of a *router*.

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 / request number
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* (*tasks*, 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

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*.

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.

router

Network element that is located between networks and guides message flows through the networks while simultaneously performing route selection, addressing and other functions. Operates on layer 3 of the OSI model.

RPC (Remote Procedure Call)

Cross-network server procedure call issued by client.

security attributes

An object's security attributes specify how and in what ways the object may be accessed.

Secure FTP

Method by which a connection is tunneled using the *FTP protocol*, thus allowing secure connections with encryption and *authentication*.

security group

Group of file attributes in the *virtual filestore*, encompassing the security attributes of a file.

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).

service class

Parameter used by *FTAM partners* to negotiate the functions to be used.

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.

SF pubset (Single Feature Pubset)

One or more disks whose key properties (disk format, allocation unit) match and which are used to store files and JVs under a shared catalog ID.

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.

Standard Access Control

Consists of the ACCESS and USER-ACCESS rights that are defined in the CREATE-FILE or MODIFY-FILE-ATTRIBUTES commands.

standard instance

The first openFT-instance that is loaded after /START-SUBSYSTEM FT. By default all openFT commands refer to this instance, if no other instance was specified with the command /SET-FT-INSTANCE. It is displayed as the first instance in the output of /SHOW-FT-INSTANCE INSTANCES=*ALL.

storage group

File attribute in the *virtual filestore*, encompasses the storage attributes of a file.

string

Character string

string significance

Describes the format of *strings* in files to be transferred using *FTAM protocols*.

subject

Active element in a data processing system from which an operation such as read, write, execute etc. can be initiated, that can cause a flow of information or can change the system status, e.g. ID, program, program component.

subsystem

Part of a system which processes a self-contained group of functions.

synchronous request

The user task that submitted the *FT request* waits for transfer to terminate. The user cannot continue working (see also *asynchronous request*).

SYSFILE environment

System files; the SYSFILE environment designates the totality of the system files assigned to a request.

system

see *FT- system*

system, local

see *local system*

system, remote

see *remote system*

system administration

– Structural unit in the computer center
– Group of individuals who employ user IDs that are associated with global privileges.

system administrator command

Command which cannot be submitted by any user ID but only by user IDs which possess the corresponding global privileges or by the TSOS user ID.

system administrator privileges

see *global privileges*

system files

The system input/output files assigned to a request. Users can only access system files indirectly by means of the SYSFILE command. System files provide data and resources that are required for the functions of the control program. System files and their primary allocations:

- SYSOUT: output of system messages to terminals
- SYSLST: output of compilation logs etc.via printer (automatic SPOOLOUT)
- SYSLSTnn: as SYSLST; 1 ≤ nn ≤ 99; each of the max.99 system files must be assigned to a cataloged file
- SYSOPT: output file as SYSLST
- SYSCMD: used to submit commands to the control program
- SYSDTA: used to enter data or statements

system resources

Resources in a computer system that can be requested or released by a *job* or a *task*.

task

Entity responsible for processes. In BS2000 tasks are used, among other things, to process user jobs (e.g. batch jobs, interactive jobs), see *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*.

transfer unit

In an FTAM environment, the smallest data unit for transporting file contents. For *FTAM-1* and *FTAM-3* these are *strings*. A transfer unit can, but need not, correspond to one file record.

Transmission Control Protocol / Internet Protocol

see *TCP/IP*

TranSON

TranSON is a software product that permits secure access to a server. The use of TranSON is transparent to the application. The connection to the remote partner goes from the workstation through a client proxy and server proxy to the remote partner. The client proxy is located on the workstation, and the server proxy is located on the remote partner. The data transferred between the client proxy and the server proxy is encrypted.

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.

TSN (Task Sequence Number)

Identification of a BS2000 process (*task*).

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.

universal-class-number

Parameter of the *document-type* that defines the *character-repertoire* of a file to be transferred.

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

Represented by a *user ID*. The term “user” is a synonym for individuals, applications, procedures etc. which can obtain access to the operating system via a user ID.

user administration

see *global user administration*

user attributes

All the characteristics of the *user ID* that are stored in the *joinfile*.

user command

Command that can be issued under any *user identification* in system mode (/) or in program mode by means of a CMD macro.

user identification / user ID

A name with a maximum length of eight characters which is entered in the joinfile. The user ID identifies the user when accessing the system. All files and job variables are set up under a user ID. The names of the files and job variables are stored in the *file catalog* together with the user ID.

user privileges

All the attributes that represent rights that are assigned to a *user identification* and are stored in the *joinfile*.

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).

virtual filestore

The FTAM virtual filestore is used by *FT systems* acting as *responders* to make their files available to their *partner systems*. The way a file is represented in the virtual filestore is defined in the FTAM standard, see *file attributes*.

VisibleString

Character repertoire for files transferred to and from *FTAM partners*.

volume set

Component of an SM pubset. A volume set is a set of disks whose key properties (disk format, allocation unit) match.
The name of the volume set is administered in a directory of the SM pubset. However, the data on a volume in the volume set is addressed via the SM pubset ID.

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

ACSE	Association Control Service Element
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Programming Interface
API/CS	Application Programming Interface/Communication System
APPC	Advanced Program-to-Program Communication
APPN	Advanced Peer-to-Peer Networking
ARP	Address Resolution Protocol
ASCII	American Standard Code for Information Interchange
ASECO	Advanced SEcurity COntrol (BS2000, SINIX)
ASN	Abstract Syntax Notation
ATM	Asynchronous Transfer Mode
BCAM	Basic Communication Access Method
BSI	Bundesamt für Sicherheit in der Informationstechnik
CAE	Common Application Environment
CCP	Communication Control Program
CCS	Coded Character Set
CCSN	Coded Character Set Name
CDDI	Copper Distributed Data Interface
CEN	Comité Européen de Coordination des Normes
CENELEC	Comité Européen de Normalisation Electrotechnique
CICS	Customer Information Control System (IBM)
CMX	Communication Manager SINIX
COM	Communication Port (asynchronous)
CPX	Compact Packet Exchange
DAS	Data Access Service
DAP	Directory Access Protocol

DBA	Data Base Access Service
DCAM	Data Communication Access Method
DCE	Data Communication Equipment
DCE	Distributed Computing Environment (OSF)
DCM	Data Communication Method
DDV	Datendirektverbindung (früher HfD)
DES	Data Encryption Standard (NBS)
DFR	Document File Retrieval
DFS	Distributed File System (DCE)
DIN	Deutsches Institut für Normung
DME	Distributed Management Environment
DMS	Data Management Service
DNS	Domain Name Service
DOS	Disk Operating System
DSA	Directory System Agent
DSC	Data Stream Compatibility
DSM	Distributed Systems Management
DSP	Directory System Protocol
DSS	Datensichtstation
DSSM	Dynamic Subsystem Management
DTE	Data Termination Equipment
DTS	Distributed Time Service
DVA	Datenverarbeitungsanlage
DVS	Datenverwaltungssystem
EBCDIC	Extended Binary-Coded Decimal Interchange Code
EMDS	Emulation Datensichtstation
EN	European Norm
ENV	Europäischer Normen-Vorschlag
EPHOS	European Procurement Handbook for Open Systems
ERMS	Entity Relationship Management System
ES	End System
ETSI	European Telecommunication Standards Institute
EWOS	European Workshop for Open Systems

FADU	File Access Data Unit
FDDI	Fiber Distributed Data Interface
FEP	Front End Processor
FJAM	File Job Access Method
FT	File Transfer
FTAC	File Transfer Access Control
FTAM	File Transfer, Access and Management (ISO 8571)
FTP	File Transfer Protocol
GOSIP	Government OSI Profile
HDLC	High Level Data Link Control (ISO 7776)
HNC	Highspeed Net Connect
HPFS	High Performance File System
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines Corporation
ICC	Intelligent Communication Controller
ICMP	Internet Control Message Protokoll
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
IGMP	Internet Group Management Protocol
IMS	Information Management System (IBM)
IP	Internet Protocol
ISAM	Index Sequential Access Method
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
IT	Information Technology
ITSEC	Information Technology Security Evaluation Criteria (Europe, White Book)
ITU	International Telecommunication Union
JCL	Job Control Language
LAN	Local Area Network
LMS	Library Maintenance System
LU	Logical Unit
MAC	Medium Access Control
MAN	Metropolitan Area Network

MCR	Magnetic Card Reader
MIB	Management Information Base
MLC	Modular LAN Connect
MSV	Mittelschnelles Synchron Verfahren
MVS	Multiple Virtual System
NCP	Network Control Program (SNA)
NCS	Network Control System
NDMS	Network Data Management System
NEA	(Name der TRANSDATA-Architektur von Siemens)
NFS	Network File System
NTP	Network Time Protocol
ODI	Open Data Link Interface
ODI	Open Device Interface
ODL	Object Description Language
OSI	Open Systems Interconnection
OSS	OSI Session Service
PAM	Primary Access Method
PC	Personal Computer
PDN	Programmsystem für Datenübertragung und Netzsteuerung
PDU	Protocol Data Unit
PEM	Privacy Enhanced Mail
PICS	Protocol Implementation Conformance Statement
PIN	Personal Identification Number
PKCS	Public Key Cryptography Standards
PLAM	Primary Library Access Method
POP	Post Office Protocol
POSIX	Portable Operating System Interface for Open Systems
PSDN	Packet Switched Data Network
PU	Physical Unit
RFC	Request for Comments
RFC1006	Request for Comments 1006
RJE	Remote Job Entry
RPC	Remote Procedure Call

RTS	Reliable Transfer Service
SAM	Sequential Access Method
SAP	Server Advertising Protocol (NetWare)
SAP	Service Access Point (OSI)
SBS	Siemens Business Services
SCM	Software Configuration Management
SDF	System Dialog Facility
SDLC	Synchronous Data Link Control
SESAM	System zur Elektronischen Speicherung Alphanumerischer Merkmale
SMTP	Simple Mail Transfer Protocol
SNA	Systems Network Architecture
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol / Internet Protocol
TELNET	Telecommunications Network Protocol
TFTP	Trivial File Transfer Protocol
TID	Transport Identification
TS	Transport System
UDP	User Datagram Protocol
UDS	Universelles Datenbanksystem
URL	Uniform Resource Locator
UTM	Universal Transaction Monitor
VDE	Verband deutscher Elektrotechniker
WAN	Wide Area Network
WS	Workstation
XDR	External Data Representation
XDS	API to Directory Service

Related publications

You will find the manuals on the internet at *http://manuals.ts.fujitsu.com*. You can order manuals which are also available in printed form at *http://manualshop.ts.fujitsu.com*.

openFT V12.0 for BS2000/OSD
Installation and Administration
System Administrator Guide

openFT V12.0 for BS2000/OSD
Program Interfaces
Programmer Reference Guide

openFT V12.0 for Unix Systems
Managed File Transfer in the Open World
User Guide

openFT V12.0 for Unix Systems
Installation and Administration
System Administrator Guide

openFT V12.0 for Windows Systems
Managed File Transfer in the Open World
User Guide

openFT V12.0 for Windows Systems
Installation and Administration
System Administrator Guide

openFT V12.0 for Unix Systems and Windows Systems
Program Interface
User Guide

openFT V12.0 for Unix Systems and Windows Systems
openFT-Script Interface
User Guide)

openFT V12.0 for z/OS
Managed File Transfer in the Open World
User Guide

openFT V12.0 for z/OS
Installation and Administration
System Administrator Guide

openNet Server (BS2000/OSD)
BCAM
User Guide

SNMP Management
SNMP Management for BS2000/OSD
User Guide

BS2000/OSD-BC
Commands (multiple volumes)
User Guide

BS2000/OSD
Executive Macros
User Guide

IMON (BS2000/OSD)
Installation Monitor
User Guide

BS2000/OSD-BC
Introductory Guide to DMS
User Guide

BS2000/OSD-BC
Subsystem Management (DSSM/SSCM)
User Guide

BS2000/OSD-BC
System Installation
User Guide

BS2000/OSD-BC
Introductory Guide to Systems Support
User Guide

JV (BS2000/OSD)
Job Variables
User Guide

SECOS (BS2000/OSD
Security Control System
User Guide

XHCS (BS2000/OSD)
8-Bit Code and Unicode Support in BS2000/OSD
User Guide

HIPLEX AF (BS2000/OSD)
High availability of applications in BS2000/OSD
Product manual

Index

%ELEMNAME 374, 389
  variable 99
%ELEMTYP 374, 389
  variable 99
%ELEMVERS 374, 389
  variable 99
%FILENAME 374, 389
  variable 99
%JOBCLASS 374, 389
  variable 99
%PARTNER 374, 389
  variable 99
%PARTNERAT
  variable 99
%RESULT 374, 389
  variable 99
%TEMPFILE 96, 369, 380
*DELETE (follow-up processing) 100, 388
*DIRECTORY
  operand description (display log records) 290
*FILE-PROCESSING
  operand description (modify profile) 241
*ftmonitor
  file name prefix 174, 234
*LOCKED
  request status 265
*MODIFY-FILE-ATTRIBUTES
  operand description (modify profile) 241
*READ-DIRECTORY
  operand description (modify profile) 241
*SUSPEND
  request status 265
*TRANSFER-FILE
  operand description (modify profile) 241
*WAIT
  request status 265

2038 139

A

abbreviate
  commands 133
abbreviated forms 133
abort file transfer
  with time specification 398
access
  to the remote system 186, 195, 199
access admission 469
access check
  FTAC 45
access control 108, 469
access protection 42, 469
  BS2000 93
  Unix system 92
  Windows 92
access right 469
access rights
  of the destination file 393
access to file
  password 204
ACCESS-MODE
  operand description (asynchronous transfer) 393
  operand description (modify remote file attributes) 252

ACCESS-RIGHTS
output field 353
ACCOUNT
operand description (asynchronous transfer) 372, 373, 386, 387
operand description (create profile) 170, 176
operand description (create remote directory) 187
operand description (delete remote directory) 193
operand description (delete remote file) 196
operand description (display remote file attributes) 350
operand description (execute remote command) 199
operand description (modify profile) 230, 237
operand description (modify remote directory) 246
operand description (modify remote file attributes) 250
account number 372, 393
FTAM 251
in the remote system 187, 386, 387
ACL (Access Control List) 92
ACT
explanation for output 334
request status 271
action list 469
ACTIVE
request status 265
ADDRESS
explanation for output 336
addressing
partner processor 84
via Application Entity Title (AET) 126
addressing options
Internet host name 85
ADEAC
explanation for output 334
ADM log records
display 289
ADM partner 85
ADM-CLIM
display setup 324
administering
files (file management) 103
admission
access, remote system 186
admission profile 47, 195, 470
create 163
create (example) 182
CSV output format 462
define directory 186
delete 188
deleting 46
display 337
locking 46
modify 219
modify (example) 242
modify privilege 226
modifying 46
name specification 166
privileged 167, 470, 484
priviliging 46
time stamp 219
admission set 44, 214, 470
basic functions 167
CSV output format 446
display 277
display (example) 279
information on (example) 217
modify 213
privileged 470, 484
ADM-LOG 325
ADM-PORT 325
ADM-TRAP-SERVER 326
Advanced Encryption Standard (AES) 470
AES 298
AES (Advanced Encryption Standard) 470
AES/RSA 48, 94
AET (Application Entity Title) 470
alias 138
alphanumeric 470
alphanum-name (data type) 139
ANSI code 470
API (Application Program Interface) 470
Application Entity Qualifier (AEQ) 126, 127
Application Entity Title (AET) 470

Application Layer 25
Application Process Title (APT) 126
Application Program Interface (API) 470
asynchronous request 32, 471
ATM
output field 355
audit 471
personal 483
authentication 52, 471
authentication check 324
authorization
access, remote system 385
login 481
LOGON 481
provide, access 93
automatic restart 34
automation 37
availability
file (FTAM) 251
of the destination file 392

B

basic functions 471, 478
admission set 167
limit (IGNORE-MAX-LEVELS) 167, 226
set (MAX-LEVELS) 215
BCAM processor name 85
binary file 205
binary format 70
binary transfer 72, 80
blank line expansion 70
BS2000
access protection 93
file types 65
BS2000 file name
(DVS) syntax 58
(POSIX) syntax 59
BS2000 host 88
BYTECNT
output description 274
BYTE-COUNT
output description 272

C

CANCEL
operand description (asynchronous transfer) 398
output description 274
cancel
FT request 156
requests (example) 161
CANCEL-FILE-TRANSFER 156
cancel file transfer 156
CANCELLED
request status 265
cat (suffix for data type) 151
cat-id (data type) 139
CCS 77
CCSN
output description 276
chaining of files, example 408
change date
accept 401
CHANGE-ATTRIBUTES
operand description (modify local file attributes) 204
character code
define 396
character repertoire 471
Character Separated Value (CSV) 471
CHARACTER-SET
operand description (modify local file attributes) 205
CHARACTERSET
output field 352
client 471
CMD
operand description (execute remote command) 198
code conversion 396
Coded Character Set (CCS) 77
CODED-CHARACTER-SET
operand description (asynchronous transfer) 377, 391
operand description (execute remote command) 199, 200

collection of files
example 404
Comma Separated Value (CSV) 471
command
abbreviate 133
execute remote 197
syntax representation 135
command execution
remote 36
with postprocessing 38
command return code 153
maincode 153
subcode1 153
subcode2 153
command-rest (data type) 139
commands
remote execution 36
communication computer 472
communication controller 472, 484
compl (suffix for data type) 146
composed-name (data type) 139
COMPRESS
operand description (asynchronous transfer) 395
output description 274
compressed transfer 31
compression 94, 472
computer network
open 472, 482
concurrency control 472
CONDITION
operand description (asynchronous transfer) 377
configuration user ID 472
connection
establishing with FTP 89
CONNECTION-LIMIT
display setup 323
connectivity 472
constraint set 71, 472
constructor (string) 149
contents type 473
control
requests issued locally 309
convert
to default admission profile 224
corr (suffix for data type) 151, 152
CP1252 21
CRE
output field 355
create
admission profile 163
default admission profile 166
remote directory 185
CREATE-FT-PROFILE 163
CREATE-PASSWORD
operand description (asynchronous transfer) 394
CREATE-REMOTE-DIR 185
creating files
in BS2000 66
CREATION-TIME
operand description (display log records) 287
CRLF (Carriage Return Line Feed) 205
cross-domain connection 473
c-string (data type) 139
CSV format
Date data type 440
Number data type 440
String data type 440
Time data type 440
CSV output
for admission sets 446
CSV output format 41
admission profile 462
admission set 446
for file attributes 467
log record 448
monitoring values 451
operating parameters 455
partner 460, 466

D

DASD (Direct Access Storage Device) 473
DATA
output description 274
data 473

data communication system 473
data compression 473
data conversion 21
data encoding 473
Data Encryption Standard (DES) 473
data integrity 399
Data Link Layer 25
data protection 473
data security 473
data transfer
  POSIX file 65
data type
  alphanum-name 139
  cat-id 139
  command-rest 139
  composed-name 139
  c-string 139
  date 139
  device 139
  filename 140
  fixed 140
  integer 142
  name 142
  partial-name 143
  posix-filename 143
  posix-pathname 143
  product-version 144
  structured-name 144
  text 144
  time 144
  vsn 144
  x-string 145
  x-text 145
data types in SDF 135, 139
  suffixes 136
DATA-ENCRYPTION
  operand description (asynchronous transfer) 399
  operand description (create profile) 181
  operand description (execute remote command) 201
  operand description (modify profile) 241
DATA-TYPE
  operand description (asynchronous transfer) 396
  operand description (execute remote command) 200
  operand description (modify local file attributes) 205
Date
  data type in CSV format 440
date (data type) 139
date and time of last modification 107
Date and time of last modification of file contents
  output field 354
DDICHK 298
DEACT
  explanation for output 334
default admission profile
  converting to 224
  creating 166
default admission set 277, 280
default for FTP 325
default for remote administration 325
default instance 255
default value 133
define
  character code 396
  direction of transfer 368
  library in admission profile 234
  local system 361
  permitted access methods 252
  permitted file access (FTAM) 204
  remote system 361
definition
  instance 25
  layer 25
  profile 25
  protocol 24
  service 25
delete
  admission profile 188
  FT request 156
  log record 51
  remote files 194
  requests (example) 161

delete remote directory 191
DELETE-FILE
operand description (modify local file attributes) 204
DELETE-FT-PROFILE 188
DELETE-REMOTE-DIR 191
DELETE-REMOTE-FILE 194
DEL-LOG 326
DENCR 298
DES 298
DES (Data Encryption Standard) 473
DES/RSA 94
description
long output 296
syntax representation 135
device (data type) 139
DICHECK
output description 275
DICHK 298
DIERR
explanation for output 334
DIR
output description 272
directory 474
create remote 185
delete remote 191
modify attributes 244
DIRECTORY-NAME
operand description (create remote directory) 185
operand description (delete remote directory) 191
operand description (modify remote directory) 244
display
admission profile 337
admission sets 277
admission sets (example) 279
AMD log records 289
FT partners (example) 333
FT profile (example) 342
FTAC logging records 284
local FT file attributes 256
log records 284
logging records (example) 302
MAX-ADM-LEVELS 279
MAX-USER-LEVELS 279
offline log files 284
openFT instance 282
operating parameter 317
operating parameters (example) 323
partner systems 328, 344
partner systems (example) 346
remote file attributes 347
display request
global request identification 266, 291
distribution
of files (example) 405
DNS name 85
document type 71, 474
dynamic partner 324
dynamic partners 84
DYN-PART
display setup 324

E

EBCDIC 21
effects
FT profile 47
ELEMENT
operand description (asynchronous transfer) 370, 382
operand description (cancel request) 159
operand description (create profile) 174
operand description (modify profile) 234
operand description (modify request queue) 210
operand description (query request status) 263
emulation 474
ENC-MAND 325
ENCR 298
ENCRYPT
output description 274
encrypted file transfer 94

encryption 48
force 181
old FT versions 49
reject 181
request description data 49
user data 49
end of file
extend 396
enter
file name 56
entering a file name
specify 56
entity 474, 479
ERASE-DATA-UNIT
operand description (modify local file attributes) 204
example
cancel requests 161
chaining of files 408
collection of files 404
create admission profile 182
display FT partners 333
display FT profile 342
display local FT file attributes 259
display logging records 302
display openFT instances 283
display operating parameters 323
display partner systems 346
display remote file attributes 358
distribution of files 405
file transfer 409
file transfer for Unix system 410
file transfer using FTAC 411
file transfer with follow-up processing 411
file transfer with password protection 403
follow-up processing 410
FTINFO for remote pre-processing 413
information on admission sets 217
job processing 407
job transfer 407
local file processing 412
local pre-processing and remote post-processing 413
long output form 296
modify remote file attributes 254
query status of file transfer request 271
remote pre-processing 413
short output form of FT logging records 294
time-driven file transfer 412
transfer of POSIX files 409
execute
remote command 197
EXECUTE-REMOTE-CMD 197
execution
password 371
exit routine 379
EXPANSION
admission profile 173
EXPIRATION-DATE
operand description (modify profile) 225, 226
explanation
OPENFT-STD (output) 322
extend of a SAM file 396
EXTEND-FILE
operand description (modify local file attributes) 204

F

FAILURE-PROCESSING 363
operand description (asynchronous transfer) 376, 391
operand description (create profile) 178
operand description (modify profile) 239
FILE
operand description (delete remote file) 194
operand description (display remote file attributes) 348
operand description (modify remote file attributes) 249
operand description (modify request queue) 209
output description 276
file
administering 103
creating (BS2000) 66
deleting 43
encrypted transfer 94
renaming 43

file (cont.)
temporary, transfer 73
transfer asynchronously 361
transfer synchronously 414
file attributes 474
CSV output format 467
display 347
display local (example) 259
modify remote 247
modify remote (example) 254
modifying 43
showing 43
file availability 108
file catalog 474
File creator
output field 353
file directory 474
file format
transparent 74
file management 35, 103, 475
description 103
FTAM attributes 104
interplay 43
local system 104
remote system 103
file management function
modify in admission profile 240
file name 47, 203
specify 47
file name prefix
*ftmonitor 174, 234
file password 64
file processing 475
file transfer
encrypted 48
example 409
for Unix system (example) 410
password protected file (example) 403
specify priority 397
using FTAC (example) 411
with follow-up processing (example) 411
with postprocessing 483
file transfer request 475
query status (example) 271
start with time specification 397
file transfer request status
query 260
File Transfer, Access and Management 477
file type
BS2000 65
FTAM 71
Unix system 69
Windows 69
z/OS 68
FILE-AVAILABILITY
operand description (asynchronous transfer) 392
operand description (modify remote file attributes) 251
output field 355
FILE-NAME
operand description (asynchronous transfer) 368, 378
operand description (cancel request) 158
operand description (create profile) 173
operand description (display local FT file attributes) 256
operand description (display log records) 290
operand description (execute remote command) 200
operand description (modify local file attributes) 203
operand description (modify profile) 233
operand description (query request status) 263
output description 272
selection criteria for canceling 158
FILENAME
output field 354
filename (data type) 140
FILE-PASSWORD
operand description (create profile) 175
operand description (modify profile) 235

files
delete remote 194
FILE-SIZE
output field 354
filesize 108
filestore 475
FIN
output description 271
FINISHED
request status 265
firewall processor 475
fixed (data type) 140
fixed-length record 475
follow-up processing 37, 38, 100, 177, 178, 236, 238, 239, 362, 475
%ELEMNAME 99
%ELEMTYP 99
%ELEMVERS 99
%FILENAME 99
%JOBCLASS 99
%PARTNER 99
%PARTNERAT 99
%RESULT 99
example 410
in the local system 374, 376
in the remote system 388, 390
instance 54
job variable 388
job variables 374
logging 375, 390
maximum length 100
overview 99
set-up 390
temporary file 374, 388
trace 375, 390
use of variables 374, 389
user ID 176
variables 99
with FTAM partners 100
follow-up processing request 475
front-end processor 474
FT
operand description (display log records) 288
FT administrator 476
FT file attributes
display 256
modify local 202
FT log record 50
FT logging
display setup 325
FT logging record
short output form (example) 294
FT logging records 284
FT profile 44
effects 47
FT request 476, 486
cancel 156
confirmation 402
delete 156
FT system 476
FT trace 476
FTAC
admission profile (privileged) 167
define directory name 192
define file name 195
fixed directory name 186
operand description (display log records) 288
password 213, 215
FTAC (File Transfer Access Control) 476
FTAC administrator 48, 476
with TSOS privilege 163
FTAC function 42
FTAC functionality 476
FTAC log record 50
FTAC logging
display setup 324
FTAC logging function 476
FTAC logging record 284
display 284
FTAC transfer admission
for FTP access 90
ftadm
protocol prefix 85
FTADM protocol 85

FTAM 27, 477
define permitted file access 204
file types 71
kernel group 106
partner 392
security group 106
specific operands 392
storage group 106
virtual filestore 106
FTAM attributes
kernel group 107
security group 108
storage group 107
FTAM catalog 109
FTAM file attributes 476
FTAM partner 27, 71, 477
addressing 85
file management 103, 104
follow-up processing 100
FTAM partners
function restrictions 392
FTAM protocol 477
FTAM standards
in openFT 27
FTAM-1 71, 474
FTAM-3 71, 474
FTAM-APPL 325
FT-FUNCTION
operand description (create profile) 180
operand description (modify profile) 240
FTINFO for remote pre-processing, example 413
FTP 26
inactive, display 325
inbound access via default FTP 89
FTP partner
addressing 85
FTP-PORT 325
functional standard 477
functionality
of layer 25
functionality class 477
future filesize 108
FUTURE-FILE-SIZE
operand description (modify remote file attributes) 251

G

gateway 477
gateway processor 477
gen (suffix for data type) 151
general string 477
GeneralString 71, 471
global privileges 478
global request identification 274
display request 266, 291
global user administration 478
GLOB-ID
output description 274
GraphicString 71, 471, 478
guard 478
GUARDS (Generally Usable Access Control Administration System) 478

H

handling
receive file 92
heterogeneous
computer systems 21
link 55
network 24, 478
HOLD
output description 271
request status 265
homogeneous link 55
homogeneous network 24, 478
host 478
HOST-NAME 325

I

IA5String 71, 471, 478
IBF 217, 279
IBP 217, 279
IBR 217, 279
IBS 217, 279

IDENTIFICATION
Einstellung anzeigen 325
explanation for output 336
identification 479
of a FT request 402
identify request 402
IDREJ
explanation for output 335
IGNORE-MAX-LEVELS
operand description (create profile) 167
operand description (modify profile) 226
inbound
file management 43, 479
follow-up processing 43, 479
receive 43, 479
request 479
requests 31
send 43, 479
submission 479
inbound access
FTP 89
inbound file management 170, 229
inbound follow-up processing 169
inbound mapping
FTAM attributes 109
inbound processing 229
inbound receive 169, 216, 228
inbound send 169, 216, 228
INBOUND-FILEMANAGEMENT 280
INBOUND-MANAGEMENT
operand description (create profile) 170
operand description (modify admission set) 217
operand description (modify profile) 229
INBOUND-PROCESSING 280
operand description (create profile) 169
operand description (modify admission set) 217
operand description (modify profile) 229
INBOUND-RECEIVE 280
operand description (create profile) 169
operand description (modify admission set) 216
operand description (modify profile) 228
INBOUND-SEND 280
operand description (create profile) 169
operand description (modify admission set) 216
operand description (modify profile) 228
index
constructor 149
global index 149
notation 150
wildcard-specific 149
INFORMATION
operand description (display local FT file attributes) 257
operand description (display log records) 293
operand description (display partners) 330
operand description (display profiles) 339
operand description (display remote file attributes) 350
operand description (query request status) 266
operand description (showing monitoring data) 304
information
getting on operating parameters 303
INI, output description 271
INITIATOR
operand description (cancel request) 158
operand description (create profile) 171
operand description (display log records) 289
operand description (modify profile) 231
operand description (query request status) 262
output description 274
initiator 479
INSERT-DATA-UNIT
operand description (modify local file attributes) 204
instance 54, 479, 482
definition 25
preprocessing, postprocessing follow-up processing 54
setting 54

instance ID 479
instance identification 52
INSTANCES
operand description (display openFT instances) 282
integer (data type) 142
integrity 479
Internet host name
addressing options 85
Internet Protocol (IP) 491
interoperability 479
interplay
file management 43
IPv4 address 85
IPv6 address 85
ISAM file
transferring to a foreign system 75
ISO 8571 27
ISO 8859 21
ISO reference model 479
ISO/IEC ISP 10607-3 27
ISO/IEC ISP 10607-6 27
ISO/OSI protocols 27
ISO/OSI reference model 479

J

JBCLJOB 374, 388
job 479
transfer 480
job class 480
JBCLJOB 374, 388
job processing
example 407
job transfer
example 407
job variable 159, 210, 264, 377
BS2000 40
contents 101
follow-up processing 374, 388
joinfile 480
JV-PASSWORD
operand description (asynchronous transfer) 377
operand description (cancel request) 160
operand description (modify request queue) 210
operand description (query request status) 264

K

kernel group 71, 107, 476, 480
attributes 107
FTAM 106
key pair set 52
KEY-LEN
display setup 324
keyword
form 134
operands 133

L

LAN (Local Area Network) 480
LAUTH 298
explanation for output 334
LAUTH2 298
layer
definition 25
LAYOUT
operand description (display admission sets) 278
operand description (display local FT file attributes) 257
operand description (display log records) 293
operand description (display operating parameters) 318
operand description (display partners (FTAC) 345
operand description (display partners) 329
operand description (display profiles) 339
operand description (display remote file attributes) 351
operand description (query request status) 266
operand description (showing monitoring data) 305
legal qualifications 108

LEGAL-QUALIFICATION
operand description (asynchronous transfer) 394
operand description (modify remote file attributes) 253
output field 356
length
RSA key 324
LIBRARY
operand description (asynchronous transfer) 370, 381
operand description (cancel request) 159
operand description (create profile) 174
operand description (modify profile) 234
operand description (modify request queue) 209
operand description (query request status) 263
library 234, 480
define in admission profile 234
name in the local system 370
name in the remote system 381
library member 480
name in the remote system 381
transferring 67
unrestricted access 174, 235
library name 234
admission profile 174
LIBRARY-ELEMENT
operand description (create profile) 174
lifetime
request 32
limit
basic functions (IGNORE-MAX-LEVELS) 167
limit basic functions (IGNORE-MAX-LEVELS) 226
link
heterogeneous 55
homogeneous 55
LISTING
operand description (asynchronous transfer) 376
LOC
explanation for output 335
output description 276
Local Area Network (LAN) 480
local file processing, example 412
local FT file attributes
display (example) 259
local pre-processing and remote post-processing, example 413
local system 368, 480
file management 104
LOCAL-PARAMETER
definition of local system 361
operand description (asynchronous transfer) 368
LOCK
output description 271
log files
output names 293
log record
display 284
log records 480
CSV output format 448
repeat output 292
logging 50
display setup 325
follow-up processing 390
postprocessing 51
preprocessing 51
logging function 480
LOGGING-ID
operand description (display log records) 286
Logical Unit (LU) 480
login
FTP 90
login admission 44
login authorization 481
LOGON authorization 170, 229, 481
long form 133
long output
description 296
long output form
example 296

low (suffix for data type) 146
LU (logical unit) 480
LUNK
explanation for output 334

M
maincode
command return code 153
mainframe 481
man (suffix for data type) 151, 152
managed file transfer 19
mandatory (suffix for data type) 152
mandatory parameter 362
MAX-ADM-LEVELS 215, 217, 279
description of output fields 279
MAX-FILESIZE
output field 356
maximum
lifetime of a request 324
number of asynchronous administration requests 324
number of connections 323
number of FT requests 324
number of tasks 323
maximum record length 91
maximum string length 71
maximum-string-length 481
MAX-LEVELS
operand description (modify admission set) 215
MAX-PARTNER-LEVEL
operand description (create profile) 172
operand description (modify profile)F) 232
MAX-REQUEST-LIFETIME 324
display setup 324
MAX-USER-LEVELS 215, 217, 279
description of output fields 279
metasyntax 137
of SDF 135
MOD
output field 355
modify
admission profile 219
admission profile (example) 242
admission set 213
file management function in admission profile 240
local FT file attributes 202
privilege in admission profile 226
remote directory attributes 244
remote file attributes 247
remote file attributes (example) 254
request queue 207
MODIFY-FILE-FT-ATTRIBUTES 202
MODIFY-FILE-TRANSFER 207
MODIFY-FT-ADMISSION-SET 213
MODIFY-FT-PROFILE 219
MODIFY-REMOTE-DIR-ATTR 244
MODIFY-REMOTE-FILE-ATTRIBUTES 247
monitoring
deactivated for partners 309
profile for 174, 234
showing setting 327
monitoring data
show 303
monitoring using job variables 101
MONJV
operand description (asynchronous transfer) 377
operand description (cancel request) 159
operand description (modify request queue) 210
operand description (query request status) 264

N
NAME
explanation for output 334
operand description (create profile) 166
operand description (delete profile) 188
operand description (display profiles) 338
operand description (display remote file attributes) 349
operand description (modify profile) 222
operand description (set an openFT instance) 255
operand description (showing monitoring data) 303

name
remote system 361
specification for admission profile 166
name (data type) 142
NCP (Network Control Program) 481
NEA 481
network
definition 24
heterogeneous 24, 478
homogeneous 24, 478
Network Control Program (NCP) 481
network description file 481
Network Layer 25
network management 24
Network Management Kernel 481
networks
openFT support 24
NEW-NAME
operand description (modify profile) 224
operand description (modify remote directory) 246
operand description (modify remote file attributes) 250
NEW-PASSWORD
operand description (modify admission set) 215
NOCON
explanation for output 334
NOKEY
explanation for output 335
non-openFT-BS2000 system 80
notational conventions 14
notational conventions for SDF 135
NUMBER
operand description (display log records) 292
Number
data type in CSV format 440
number
display maximum of transport connections 323

O

object 481
OBR 217, 279
OBS 217, 279
odd (suffix for data type) 151
offline log file
selection according to date 291
selection according to name 291
specify number 292
offline log files
display 284
offline log records
view 291
old FT versions
encryption 49
open computer network 472
openEdition file 68
syntax 63
openFT
partner 482
openFT add-on products 23
openFT instance
display 282
display (example) 283
set 255
openFT instances 54
openFT partner 27
addressing 85
file management 103
openFT protocol
addressing with 85
openFT protocols 27, 482
openFT-AC 219
openFT-AC for BS2000 115
OPENFT-APPL
display setup 325
openFT-FTAM 482
OPENFT-STD
output explanation 322
operand 133
operand value
constant 133
introductory 133

operating parameter
display 317
display (example) 323
outputting 303
operating parameters 482
CSV output format 455
OPS variables 154
SHOW-FILE-FT-ATTRIBUTES 258
SHOW-FILE-TRANSFER 268
SHOW-FT-ADMISSION-SET 281
SHOW-FT-INSTANCE 283
SHOW-FT-LOGGING-RECORDS 300
SHOW-FT-MONITOR-VALUES 306
SHOW-FT-OPTIONS 319
SHOW-FT-PARTNERS 331
SHOW-FT-PROFILE 340
SHOW-FT-RANGE 346
SHOW-REMOTE-FILE-ATTRIBUTES 357
OSI Reference Model 25
functionality 25
OSI reference model 479
outbound
receive 43, 482
request 482
requests 31
send 43, 482
submission 482
outbound encryption
activate 219
outbound receive 169, 216, 228
outbound request 207
outbound send 168, 216, 227
OUTBOUND-RECEIVE 280
operand description (create profile) 169
operand description (modify admission set) 216
operand description (modify profile) 228
OUTBOUND-SEND 280
operand description (create profile) 168
operand description (modify admission set) 216
operand description (modify profile) 227
OUTPUT
operand description (asynchronous transfer) 377
operand description (display admission sets) 278
operand description (display local FT file attributes) 257
operand description (display log records) 293
operand description (display operating parameters) 318
operand description (display partners (FTAC) 345
operand description (display partners) 329
operand description (display profiles) 339
operand description (display remote file attributes) 351
operand description (execute remote command) 200
operand description (query request status) 266
operand description (showing monitoring data) 305
output fields
description (show log record) 294
description (show operating parameters) 323
description (SHOW-REMOTE-FILE-ATTRIBUTES) 352
output fields (OPS)
meaning (show operating parameters) 322
output in CSV format 41
admission sets 446
SHOW-FILE-TRANSFER 441
SHOW-FT-LOGGING-RECORDS 448
SHOW-FT-MONITOR-VALUES 451
SHOW-FT-OPTIONS 455
SHOW-FT-PARTNERS 460
SHOW-REMOTE-FILE-ATTRIBUTES 467
overwrite
receive file 395
OWNER
output description 275

owner 483
of a FT request 156
of FT request 483
OWNER-IDENTIFICATION 158
OWNER-IDENTIFICATION
operand description (cancel request) 158
operand description (delete profile) 189
operand description (display log records) 287
operand description (display profiles) 338
operand description (modify profile) 223
operand description (modify request queue) 209
operand description (query request status) 262

P

PAM file
fetching from a foreign system 75
transferring to a foreign system 75
partial-filename (data type) 143
PARTNER
operand description (asynchronous transfer) 368
operand description (cancel request) 158
operand description (create profile) 172
operand description (create remote directory) 185
operand description (delete remote directory) 191
operand description (delete remote file) 194
operand description (display log records) 289
operand description (display partners) 329
operand description (display remote file attributes) 348
operand description (execute remote command) 198
operand description (modify profile) 232
operand description (modify remote directory) 244
operand description (modify remote file attributes) 249
operand description (modify request queue) 209
operand description (query request status) 262
output description 272, 275
partner
CSV output format 460
partner address 84
partner list 84
partner name 84
partner processor
addressing 84
partner system 361, 483
display 328, 344
display (example) 346
specify 47
PARTNER-CHECK
display setup 324
PARTNER-STATE
operand description (query request status) 262
PASSWORD
operand description (asynchronous transfer) 371, 372, 373, 384, 386, 387
operand description (create profile) 166, 171, 177
operand description (create remote directory) 186, 187
operand description (delete profile) 189
operand description (delete remote directory) 192, 193
operand description (delete remote file) 195, 196
operand description (display remote file attributes) 349, 350
operand description (execute remote command) 199
operand description (modify admission set) 214
operand description (modify local file attributes) 203
operand description (modify profile) 223, 230, 237

PASSWORD (cont.)
operand description (modify remote directory) 245, 246
operand description (modify remote file attributes) 249, 250
password 213, 277, 483
access, remote system 187
for execution 371
path-compl (suffix for data type) 146
P-CHK
explanation for output 335
PDSE member 62
permitted actions 107, 483
PERMITTED-ACTIONS
operand description (modify local file attributes) 204
Personal Audit for Individual Accountability 483
physical Layer 25
physical layer 25
PO member 62
POLLING
operand description (showing monitoring data) 304
polling
cancel (log records) 293
log records 292
polling interval
log records 292
polling log records
number of repetitions 292
port number 325, 483
default for openFT 325
partner host 85
Portable Open System Interface (POSIX) 483
positional form 134
positional operands 133
POSIX (Portable Open System Interface) 483
POSIX file
file format during transfer 65
posix filename (data type) 63
posix path name (data type) 59
posix pathname (data type) 63
POSIX wildcard 147
posix-filename (data type) 143
posix-pathname (data type) 143
post-processing
set up 163
postprocessing 37, 483
function 38
instance 54
logging 51
previous FT versions 38
post-processing command
local 369, 380
PREFIX
operand description (create profile) 178, 179
operand description (modify profile) 238, 240
prefix
specify for file name 47
specify for follow-up processing 48
pre-processing
set up 163
preprocessing 37, 38, 484
description 96
instance 54
logging 51
pre-processing command
local 369
remote system 380
preprocessor 484
presentation 484
Presentation Layer 25
presentation selector 484
partner host 87
PRIO
output description 274
PRIORITY
operand description (asynchronous transfer) 397
operand description (modify request queue) 211
priority 397
partners 33
specify for file transfer 397
priority control 33
private key 484
privilege 484

PRIVILEGED 219
operand description (create profile) 167
operand description (modify profile) 226
privileged admission profile 484
privileged admission set 470, 484
privileges
global 478
procedure call
postprocessing 38
processing
prohibited 47
specified 47
PROCESSING-ADMISSION
operand description (asynchronous transfer) 372, 386
operand description (create profile) 176
operand description (modify profile) 236
specify user ID for follow-up processing 363
PROCESS-LIMIT
display setup 323
product range
openFT 22
product-version (data type) 144
profile 484
definition 25
program call
postprocessing 38
preprocessing 38
program interfaces 40
prohibited processing 47
PROTECT
output description 275
protection attributes
transferring 95
protocol 485
definition 24
PS dataset 61
public key 485
public space 485
Public Volume Set 485
pubset 485
PW 279

Q

query
status of file transfer request 260
QUEUE-POSITION
operand description (modify request queue) 211
quotes 133
quotes (suffix for data type) 152

R

RAUTH 298
explanation for output 334
RAUTH2 298
REA
output field 355
read password
receive file 371, 384
send file 371
READ-ATTRIBUTES
operand description (modify local file attributes) 204
READ-FILE
operand description (modify local file attributes) 204
Readme file 15
REASON-CODE
operand description (display log records) 290
receive file 485
handling 92
overwrite 395
read password 371, 384
write password 371, 384
receive system 368, 485
record 485
record length 91, 475, 493
record-by-record transfer 74
RECORD-FORMAT
operand description (asynchronous transfer) 399
operand description (modify local file attributes) 205
output field 355

RECORD-SIZE
operand description (asynchronous transfer) 399
operand description (modify local file attributes) 205
output field 355
RECORD-TYPE
operand description (display log records) 288
relay 485
relay program 486
REM
explanation for output 335
output description 276
remote command execution 36
remote file attribute
display (example) 358
remote pre-processing, example 413
remote system 378, 486
file management 103
identification of user 350
REMOTE-PARAMETER
definition of remote system 361
operand description (asynchronous transfer) 378
REPLACE-FILE
operand description (modify local file attributes) 204
request 486
asynchronous 32, 471
cancel (example) 161
file management 103
identify 402
lifetime 32
owner 156
priority 33
synchronous 32, 414, 490
request description data
encrypting 20
Request for Comments (RFC) 487
request ID 486
request identification 486
request information
about FT requests 260
request lifetime 324
request management 486
request number 478, 486
request queue 33, 486
modify 207
request storage 486
REQUEST-LIMIT
display setup 324
resources 487, 491
responder 487
restart 487
automatic 34
pre- and post-processing 370
pre-/post-processing 381
restart capability
postprocessing 98
restart point 487
restriction
transfer direction 47
write mode (FT profile) 48
result list 376, 487
result message 362, 402
return code, command 153
RFC (Request for Comments) 487
RFC1006 487
RFC959 26
Rivest-Shamir-Adleman procedure 487
router 487
ROUTING
explanation for output 336
RPC (Remote Procedure Call) 487
RSA 298
RSA key, length 324
RSA procedure 487
RSA/AES 48, 94
RSA/DES 94
RUNK
explanation for output 334

S

S.FT-BS2.transfer-id.JOB 374, 388
SAM file extend 396
scope ID 85

SDF
syntax representation 135
SECLEV
explanation for output 335
Secure FTP 488
secure operation 42
security attributes 488
security group 108, 476, 488
attributes 108
FTAM 106
security level 172, 215, 488
default value 324
SECURITY-LEVEL
display setup 324
SELECT
operand description (cancel request) 157
operand description (display log records) 286
operand description (modify request queue) 208
operand description (query request status) 262
selection criteria
for file transfers to be cancelled 159
for FT requests 157
for FT requests to be canceled 157
for outbound requests to be modified 208
SELECT-PARAMETER
operand description (delete profile) 189
operand description (display partners (FTAC) 345
operand description (display profiles) 338
operand description (modify admission set) 215
operand description (modify profile) 223
send file 488
binary transfer 396
read password 371
send system 368
sending system 488
sep (suffix for data type) 151
server 488
service 488
definition 25
service class 488
session 489
Session Layer 25
session selector 489
partner computer 86
set
openFT instance 255
set up
post-processing 163
pre-processing 163
SET-FT-INSTANCE 255
setting an instance 54
setup
transfer admission 166
SF pubset 489
short form 133
show
monitoring data 303
SHOW-FILE-FT-ATTRIBUTES 256
SHOW-FILE-TRANSFER 260
output in CSV format 441
SHOW-FT-ADMISSION-SET 277
SHOW-FT-INSTANCE 282
OPS variables 283
SHOW-FT-LOGGING-RECORDS 284
SHOW-FT-MONITOR-VALUES 303
CSV format 451
OPS variables 306
SHOW-FT-OPTIONS 317
get information on monitoring data 303
SHOW-FT-PARTNERS 328
Beispiel 333
SHOW-FT-PROFILE 224, 337
example 342
SHOW-FT-RANGE 344
SHOW-REMOTE-FILE-ATTRIBUTES 347
example 358
Siemens protocols 27
Simple Network Management Protocol (SNMP) 489
Single Feature Pubset 489
SN77309 27
SN77312 27
SNA network 489

SNMP 24
SNMP (Simple Network Management Protocol) 489
special form (*DELETE) 100
specify 47
  constructor 150
  file transfer request 55
  partner processor 84
  partner systems 47
  prefix for file name 47
  prefix for follow-up processing 48
  processing 47
  syntax rules 91
  transfer admission 88
  user ID for follow-up processing 363
Standard Access Control 489
standard admission profile 47
standard admission set 489
standard entity 489
START 397
  operand description (asynchronous transfer) 397
  output description 274
start of file 395
start of the file transfer 397
STARTED
  display setup 323
STATE
  explanation for output 334
  operand description (display partners) 330
  operand description (query request status) 265
  output description 271, 274
status
  of FT request 265
storage group 107, 476, 489
  attribute 107
  FTAM 106
STORAGE-ACCOUNT
  operand description (asynchronous transfer) 393
  operand description (modify remote file attributes) 251
  output field 354
String
  data type in CSV format 440
string 489
string significance 71, 489
structured-name (data type) 144
subcode1
  command return code 153
subcode2
  command return code 153
subject 490
subsystem 490
successful file transfer 376
SUCCESS-PROCESSING 363
  operand description (asynchronous transfer) 376, 390
  operand description (create profile) 177
  operand description (modify profile) 238
SUFFIX
  operand description (create profile) 178, 179
  operand description (modify profile) 239, 240
suffixes for data types 136, 146
super FTAC administrator 163
SUSP
  output description 271
synchronous request 32, 414, 490
syntax
  BS2000 378
  BS2000 file name (DVS) 58
  BS2000 file name (POSIX) 59
  MSP 378
  Unix system file name 60
  Windows file name 60
  z/OS file name 61
syntax check 378
syntax description 135
syntax representation
  SDF 135
syntax rules
  specify 91
SYSCMD 491
SYSDTA 491
SYSFILE environment 490
SYSLST 491
SYSOPT 491

SYSOUT 491
system 490
  local 480, 490
  remote 486, 490
system administration 490
system administrator command 490
system administrator privileges 490
system files 491
system resources 491
SYSUT1 380

T
tabulator expansion 70
task 491
Task Sequence Number (TSN) 492
TCP/IP 491
Tempfile 73
temp-file (suffix for data type) 151
temporary file
  follow-up processing 388
  transfer 73
text (data type) 144
text file 205
  transfer 396
  transfer as 80
text format 69
  data conversion 21
text transfer 72
Time
  data type in CSV format 440
time (data type) 144
time stamp
  updating on admission profile 219
time-driven file transfer, example 412
Top Secret 491
TRACE
  display setup 327
  explanation for output 335
trace
  follow-up processing 375, 390
TRANS
  output description 274
TRANS-ADM
  output description 276
transfer
  as a text file 396
  binary 396
  encrypted 94
  file asynchronously 361
  file synchronously 414
  in binary format 72
  in compressed form 395
  in text format 72
  in user format 72
  record-by-record 74
  text file 80
  transparent format 74
transfer admission 223, 224, 337, 371, 491
  file transfer request 47
  FTAC 44
  setup 166
  specify 88
transfer binary 80
transfer direction 231
  restriction 47
transfer file
  DVS file 65
  file name syntax 65
  library element 65
  PLAM library 65
  POSIX file 65
  text file 80
transfer ID 157, 208, 261
transfer of POSIX files, example 409
transfer unit 491
TRANSFER-ADMISSION 219
  operand description (asynchronous transfer) 371, 385
  operand description (create profile) 166
  operand description (create remote directory) 186
  operand description (delete profile) 189
  operand description (delete remote directory) 192
  operand description (delete remote file) 195
  operand description (display profiles) 338
  operand description (display remote file attributes) 349

TRANSFER-ADMISSION (cont.)
operand description (execute remote command) 199
operand description (modify profile) 223, 224, 225
operand description (modify remote directory) 245
operand description (modify remote file attributes) 249
TRANSFER-ATTRIBUTES
operand description (modify local file attributes) 204
TRANSFER-DIRECTION
operand description (asynchronous transfer) 368
operand description (create profile) 172
operand description (modify profile) 231
TRANSFER-FILE 361
full form 363
TRANSFER-FILE-SYNCHRONOUS 414
TRANSFER-ID
operand description (cancel request) 157
operand description (modify request queue) 208
operand description (query request status) 261
output description 274
request identification 157
transferring
library members 67
TRANS-ID
output description 271
Transmission Control Protocol (TCP) 491
TRANSP
output description 274
TRANSPARENT
operand description (asynchronous transfer) 397
transparent file format 74
transparent format
transfer 74
transport connection 492
transport connections
display maximum number 323
transport layer 25, 492
transport protocol 26, 492
transport selector 492
partner host 86
transport system 26, 27, 492
transport unit
maximum size 324
TRANSPORT-UNIT-SIZE
display setup 324
TRAP
display setup 326
T-selector 492
TSN (Task Sequence Number) 492
TYPE
operand description (asynchronous transfer) 371, 383
operand description (cancel request) 159
operand description (create profile) 175
operand description (modify profile) 235
operand description (modify request queue) 210
operand description (query request status) 264
types
follow-up processing 99

U

umlauts
data conversion 21
UNC names 60
under (suffix for data type) 147
Unicode 21
universal-class-number 492
Unix system
access protection 92
file name, syntax 60
file types 69
UNIX(TM) 493
USAGE
operand description (modify profile) 225, 226
user 493
user (suffix for data type) 152
user administration 493
user attributes 493

user command 493
user data
encryption 49
user format 70
transfer 72
user ID 214, 236, 277, 493
admission profile 176
delete admission profile 189
in the remote system 187
user identification 493
in the remote system 387
user privilege 493
USER-ADMISSION
operand description (create profile) 170
operand description (modify profile) 229
USER-IDENTIFICATION
operand description (asynchronous transfer) 372, 373, 386, 387
operand description (create profile) 170, 176
operand description (create remote directory) 187
operand description (delete remote directory) 193
operand description (delete remote file) 196
operand description (display admission sets) 277
operand description (display partners (FTAC) 345
operand description (display remote file attributes) 350
operand description (execute remote command) 199
operand description (modify admission set) 214
operand description (modify profile) 230, 236
operand description (modify remote directory) 246
operand description (modify remote file attributes) 250
USER-INFORMATION
operand description (create profile) 180
operand description (modify profile) 241

V

VALID
operand description (modify profile) 225
variable-length record 493
variables
follow-up processing 99
vers (suffix for data type) 152
VERSION
operand description (asynchronous transfer) 370, 382
operand description (cancel request) 159
operand description (modify profile) 235
operand description (modify request queue) 210
operand description (query request status) 263
virtual filestore 28, 106, 494
FTAM 106
VisibleString 71, 471, 494
volume set 494
VSAM file 62
vsn (data type) 144

W

WAIT
output description 271
WAN (Wide Area Network) 494
Wide Area Network (WAN) 494
wild(n) (suffix for data type) 147
wildcards
partners in ftshwl 290
Windows
access protection 92
file types 69
Windows file name
syntax 60
with (suffix for data type) 146
without (suffix for data type) 151
without-cat (suffix for data type) 151
without-corr (suffix for data type) 151
without-gen (suffix for data type) 151
without-man (suffix for data type) 151
without-odd (suffix for data type) 151
without-sep (suffix for data type) 151

without-user (suffix for data type) 152
without-vers (suffix for data type) 152
WRITE
    output description 274
write mode
    restriction 48
write password
    receive file 371, 384
WRITE-MODE
    operand description (asynchronous transfer) 395
    operand description (create profile) 179
    operand description (modify profile) 240

X

x-string (data type) 145
x-text (data type) 145

Z

z/OS
    file name, syntax 61
    file type 68
z/OS UNIX System Services 63