English

FUJITSU Software

openFT (z/OS) V12.1

Command Interface

User Guide

Edition July 2017

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 Preface

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 offers an extensive range of file transfer products - the openFT product range - for the following system platforms:

- BS2000®
- Linux® (Intel x86 and x86_64 / IBM z Systems), Solaris™ (SPARC®/Intel™), AIX®, HP-UX®
- Microsoft® Windows™ 8.1, 10, Windows Server 2012 R2, Windows Server 2016
- z/OS (IBM®)

1.1 Brief description of the product

FUJITSU Software openFT (z/OS) is the file transfer product for computers using the operating system z/OS.

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

The range of functions made available by openFT can be extended by:

- FTAC:

  FTAC provides extended system and data access protection. FTAC stands for File Transfer Access Control.
  On z/OS, FTAC is provided by the add-on product openFT-AC.

- openFT-FTP:

  openFT also supports the FTP functionality. This makes it possible to interconnect with other FTP servers.

1.2 Target group

This manual is intended for those who want to use the command openFT interface on a z/OS in order to transfer files and to administer openFT.

1.3 Concept of openFT manuals

openFT - Concepts and Functions

This manual is intended for those who want to get familiar with the capabilities of openFT and want to understand the openFT functions. It describes:

- the concept of openFT as a Managed File Transfer
- the scope of work and main features of the openFT product family
- the openFT-specific terms

openFT (Unix and Windows Systems) - Installation and Operation

This manual is intended for the FT, FTAC and ADM administrator on Unix and Windows systems. It describes:

- how to install openFT and its optional components
- how to operate, control and monitor the FT system and the FTAC environment
- the configuration and operation of a remote administration server and a ADM trap server
- important CMX commands on Unix systems

openFT (BS2000) - Installation and Operation

This manual is intended for the FT and FTAC administrator on BS2000 systems. It describes:

- how to install openFT and its optional components on the BS2000 system
- how to operate, control and monitor the FT system and the FTAC environment
- the accounting records

openFT (z/OS) - Installation and Operation

This manual is intended for the FT and FTAC administrator on z/OS. It describes:

- how to install openFT and its optional components, including the requirements for using the product
- how to operate, control and monitor the FT system and the FTAC environment
- the openFT and openFT-AC messages for the FT administrator
- additional sources of information for the FT administrator, such as the accounting records and the logging information

openFT (Unix and Windows Systems) - Command Interface

This manual is intended for the openFT users on Unix and Windows systems and describes:

- the conventions for file transfer to computers with different operating systems
- the openFT commands on Unix and Windows systems
- the messages of the various components

The description of the openFT commands also applies to the POSIX interface on BS2000 systems.

openFT (BS2000) - Command Interface

This manual is intended for the openFT users on BS2000 systems and describes:

- the conventions for file transfer to computers with different operating systems
- the openFT commands on BS2000 systems
- the messages of the various components

openFT (z/OS) - Command Interface

This manual is intended for the openFT users on z/OS systems and describes:

- the conventions for file transfer to computers with different operating systems
- the openFT commands on z/OS
- the menu interface for the FT administrator and the FT user
- the program interface for the FT user
- the messages of the various components

openFT (BS2000) - Program Interface

This manual is intended for the openFT programmer and describes the openFT and openFT-AC program interfaces on BS2000 systems.

openFT (Unix and Windows Systems) - C and Java Program Interface

This manual is intended for C and Java programmers on Unix and Windows systems. It describes the C program interface and the main features of the Java interface.

openFT (Unix and Windows Systems) - openFT-Script Interface

This manual is intended for XML programmers and describes the XML statements for the openFT-Script interface.

i Many of the functions described in the manuals can also be executed via the openFT graphical interface, the openFT Explorer. The openFT Explorer is available on Unix systems and Windows systems. You can use the openFT Explorer to operate, control and monitor the FT system and the FTAC environment of remote openFT installations on any system platform independent from the local system, A detailed online help system that describes the operation of all the dialogs is supplied together with the openFT Explorer.

1.4 Changes since the last version

This section describes the changes in openFT V12.1 compared to openFT V12.0A.

> The functional extensions to the openFT commands, whether they relate to administrators or users, are also available in the openFT Explorer which is provided on Unix and Windows systems. For details, see the *New functions* section in the associated online help system.
>
> On z/OS, the functional extensions are also available in the menu system (panels).

1.4.1 Changes for all platforms

- Extended Unicode support

  On all Unicode capable systems, file names, FTAC transfer admissions and follow-up processing may consist of Unicode characters. To permit this, the function "Encoding Mode" has been introduced in order to represent the Unicode names correctly on all involved systems.

  The command interfaces have been extended as follows:

  – All platforms:

    The new field FNC-MODE in the long output of log records displays the encoding mode for the file name (commands *ftshwl*, SHOW-FT-LOGGING-RECORDS and FTSHWLOG). On BS2000 systems, the OPS variables have been extended by the elements FNC-MODE and FNCCS.

  – Unix systems and Windows systems:

    – New option *-fnc* in order to set the encoding mode in a file transfer, file management or administration request. This option is available for the commands *ft*, *ftadm*, *ftcredir*, *ftdel*, *ftdeldir*, *ftexec*, *ftmod*, *ftmoddir*, *ftshw* and *nopy*.
The encoding mode is displayed in the output of the following commands (in addition to *ftshwl*): *ftshw* and *ftshwr* (FNC-MODE field).
The number of not mapped file names is displayed using *ftshw -sif*.
    – New attribute *CmdMode* in the configuration of remote administration server to define the (recommended) encoding mode for administered openFT instances.
The encoding mode is displayed in the output of the *ftshwc* command (MODE field).
This function is also available in the configuration editor of the openFT Explorer.
    – In Unix systems, it is also possible to set the character set which is to be used for inbound requests in character mode. To do this, the new option *-fnccs* in the *ftmodo* command has been introduced.

The character set which is currently set for inbound requests in character mode is displayed in *ftshwo*, FN-CCS-NAME field.

- For inbound requests, the long output and CSV output of log records display the address of the partner system in the new field PTNR-ADDR. On BS2000 systems, the partner address is also displayed in the OPS variable PARTNER-ADDRESS.
- Deactivation of the restart functions

  The restart function can be deactivated for asynchronous file transfer requests via the openFT or FTAM protocol. The restart can be set partner-specifically for outbound requests and globally for inbound and outbound requests. To permit this, the following commands have been modified:

  Unix and Windows systems:
  - *ftaddptn* and *ftmodptn*: New option *-rco*
  - *ftmodo*: New options *-rco* and *-rci*

  BS2000 and z/OS systems:
  - ADD-/MODIFY-FT-PARTNER and FTADDPTN/FTMODPTN:
    New operand RECOVERY-OUTBOUND
  - MODIFY-FT-OPTIONS and FTMODOPT:
    New operands RECOVERY-OUTBOUND and RECOVERY-INBOUND
- Minimum RSA key length for openFT protocol

  An openFT instance can require a minimum RSA key length for the openFT session encryption. The minimum RSA key length can be defined in the operating parameters. To permit this, the following commands have been modified:

  Unix and Windows systems:
  - *ftmodo*: New option *-klmin*

  BS2000 and z/OS systems:
  - MODIFY-FT-OPTIONS and FTMODOPT: New parameters RSA-PROPOSED and RSA-MINIMUM for the KEY-LENGTH operand.
- Minimum AES key length for openFT protocol

  An openFT instance can require a minimum AES key length for the openFT session encryption. The minimum AES key length can be defined in the operating parameters. To permit this, the following commands have been modified:

  Unix and Windows systems:
  - *ftmodo*: New option *-aesmin*

  BS2000 and z/OS systems:
  - MODIFY-FT-OPTIONS and FTMODOPT: New parameter AES-MINIMUM for the KEY-LENGTH operand.

1.4.2 Changes for Unix and Windows platforms

- Transferring directories:
  - Directories can be transferred between Unix and Windows systems. To permit this, the commands *ft* and *ncopy* have been extended with the option *-d*.
  - The new field PROGRESS in the output of the *ftshwr* command displays the progress of (asynchronous) directory transfer.
  - The new option *ftmodo -ltd* has been introduced to set the logging scope for directory transfer.
  - The new value *ftshwl -ff=T* selects log records of directory transfer requests. In addition, the *ftshwl* output has been extended to the field TRANSFILE (long output) as well as the FT function values TD, SD, SF (short output) and the value FUNCTION=TRANSFER-DIR (long output).
- Transferring multiple files via FTAM:

  Multiple files can be transferred synchronously between Unix and Windows systems using the FTAM protocol. This is controlled by a specific file name syntax of the *ncopy* command.
- Extension of the openFT-Script commands
  - The FT administrator can set limits of openFT requests. To permit this, the command *ftmodsuo* has been extended to the options *-u*, *-thl* and *-ftl*.
  - *ftshwsuo* displays the limits currently set.
- The *ftshwk* command displays the partner name for public keys of partner systems.
- FarSync X25 support

  FarSync X.25 cards from the manufacturer FarSite are directly supported by openFT on Linux and Windows systems. PCMX is no longer required for this. The connection method XOT (X.25 via TCP/IP) is also supported on Linux by using the FarSync XOT Runtime.
  To permit this, the commands *ftaddptn*, *ftmodptn*, *ftmodo*, *ftshwptn* and *ftshwo* have been extended.
- Extended support of the Application Entity Title

  The Application Entity Title (AET) now can be used for checking the partner address of FTAM partners. To permit this, the *ftmodo* command has been modified by extending the *-ptc* (partner check) option and adding the *-aet* option for specifying the AET. The *ftshwo* command has been extended by the *-ae* option.

- Other changes
  - Modified partner checking for partners which are addressed via IPv6 with scope ID or via X.25 with line number. By this, a unique identification via the partner address is always possible.
  - The *ft_mget* command has been extended by the *-case* option which controls the consideration of the upper case / lower case in the file name pattern.
  - The ADM administrator now can return the permission for remote administration (*ftmoda -admpriv=n*). The configuration of the remote administration server is retained.

1.4.3 Changes for Unix platforms

- Single-user mode

  On Unix systems, the administrator can switch between the multi-user mode (default) and the single-user mode using the *ftsetmode* command. In single-user mode openFT runs completely under a specific user ID (the so called openFT ID) which is also FT and FTAC administrator. To permit creating and administering additional openFT instances in single-user mode, the commands *ftcrei* and *ftmodi* have been extended by the option *-ua* for specifying the user ID of a new instance.
- openFT release for Linux 64 bit.
- SNMP is no longer supported on Unix platforms.

1.4.4 Changes for BS2000 systems and z/OS

- New commands GET-REMOTE-FILES (BS2000 systems) and FTMGET (z/OS) for synchronous or asynchronous fetching of multiple files specified by wildcards from a remote system.
- New diagnostics command FTPING on BS2000-POSIX and z/OS for testing the openFT connection to a remote partner.

1.4.5 Changes for z/OS

- The PARM member of the z/OS parameter file has been changed as follows:
  - New key word JOB_JOBCLASS for follow-up processing jobs, preprocessing jobs, postprocessing jobs and print jobs.
  - New key word LISTPARM for setting of a default printer (LISTING=*STD in a FT request).
  - The key word JOB_MSGCLASS now applies to preprocessing jobs and postprocessing jobs.
- For FJBATCH in z/OS as of V2.1, you can use the PARMDD parameter instead for the PARM parameter.
- NCOPY and FTACOPY: New value LISTING=*STD in LOCAL-PARAMETER in order to assign a printer defined via LISTPARM.
- openFT (z/OS) is now supporting host names with up to 80 characters in length. This applies both to the internal communication in z/OS and to connections to z/OS partners.
- The member TNCTCPIP of the z/OS parameter file is no longer supported, therefore the description has been dropped.

1.4.6 New functions that are only available in the openFT Explorer

- Exporting public keys

  The FT administrator can export public keys of the local openFT instance using the *Key Management* - *Export Public Key* command in the *Administration* menu.
- Deleting diagnosis information and console messages

  The FT administrator can delete diagnosis information and console messages using the commands *Delete Diagnosis Information* and *Delete Console Messages* in the *Administration* menu.
- The logging is also available in the object tree of the openFT Explorer.

Please refer to the online help for more details.

1.5 Notational conventions

The following notational conventions are used throughout this manual:

`typewriter font`
typewriter font is used to identify entries and examples.

ℹ indicates notes.

⚠ Indicates warnings.

Additional conventions are used for the command descriptions, see section “Command syntax representation” on page 42.

1.6 README files

Information on any functional changes and additions to the current product version can be found in product-specific README files.

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

1.7 Current information on the Internet

Current information on the openFT family of products can be found in the internet under *http://www.fujitsu.com/ts/openFT*.

2 Menu interface

This chapter describes the easy-to-use menu interface via which you can perform your user tasks or your FT and FTAC administrator tasks. The previous distinction between one menu system for administrators and another for users has been discarded and both now see the same entry menu (Primary Option Menu). Naturally, FT and FTAC administrators have more rights.

All openFT functions are supported by the ISPF product provided that this product is installed in the z/OS system. In this case, the FT user and the FT administrator are provided with the user-friendly ISPF interface with the usual choice of menus and data entry panels, immediate warnings in the event of errors, help functions etc. When you use this menu interface, the corresponding commands are issued internally. The panels of the menu interface are described below.

A description of how to enter the commands NCOPY, NCANCEL and NSTATUS directly under TSO is given in the chapter “openFT commands” on page 97.

Notes for the FT administrator

The menu interface is a more user-friendly possibility of executing almost the same functions that can be executed via the command interface, i.e.:

- load and start the openFT load module,
- modify the request file and the partner list,
- activate the openFT instance,
- control the use of resources,
- request information about the openFT instance,
- monitor the openFT instance,
- request information on FT requests,
- cancel/abort FT requests,
- deactivate and reactivate file transfer requests submitted in the local system to individual remote FT systems,
- deactivate the openFT instance,
- terminate the openFT load module,
- administer the local keys of an openFT instance,
- administer FTAC admission sets and FTAC admission profiles if required,
- administer remote openFT instances on any platforms.

In contrast, the menu interface allows you to edit the members PARM, TSOJOB, JCLJOB, PRTJOB, TSOVVJOB, TSONVJOB, TSOVFJOB, SUCCMSG, FAILMSG and TNSTCPIP of the FT parameter library directly. The FT parameter library itself, however, must already exist.

Exceptions

- For the FJGEN command for setting the form installation parameters the menu interface does not provide a function which corresponds to this command; you must enter the command directly in order to perform the corresponding installation step. The parameters you set in this installation step, however, can be displayed at any time via a menu.
- You cannot process the member FNAMECTB in the FT parameter library via the menu interface.

Please refer to the manual "openFT (z/OS) - Installation and Operation" for details on FJGEN on the FT parameter library.

In a running TSO session, you can perform these allocations manually. This means, for example, that you can also change the openFT instance within the TSO session. You should store the necessary commands in a CLIST and execute these in TSO mode. You can also record the name of this CLIST in a LOGON procedure so that the commands are always executed when you log on.

2.1 Requirements

In order to use the menu interface, the following requirements have to be fulfilled:

- The libraries with the openFT panel definitions, CLISTs and messages is installed.
- The libraries necessary for using the menu interface is concatenated for the desired instance.

For details, see the manual "openFT (z/OS) - Installation and Operation", ... „Setting the openFT instance“.

Setting an openFT instance

The panels always apply to the currently set openFT instance. If you want to use an instance other than STD you must set the instance:

– The CLIST <openft qualifier>.<inst>.CLIST must be concatenated

– The file <openft qualifier>.<inst>.CONN must be allocated with the DD name OPENFT.

The FT administrator defines the specifications for the OPENFT QUALIFIER and the name of the instance (<inst>) when setting up the instance.

In a running TSO session, you can perform these allocations manually. This means, for example, that you can also change the openFT instance within the TSO session. You should store the necessary commands in a CLIST and execute these in TSO mode. You can also record the name of this CLIST in a LOGON procedure so that the commands are always executed when you log on.

Example of this type of CLIST with instance INST2

```
allocate file(sysproc) dataset('isp.sispclib' -
'OPENFTQU.STD.CLIST'                           -
'OPENFTQU.OPENFT.PANEL.CLIST' ) reuse shr
allocate file(ispplib) dataset('isp.sisppenu' -
’OPENFTQU.OPENFT.PANELS') reuse shr
allocate file(ispmlib) dataset('isp.sispmenu' -
'OPENFTQU.OPENFT.PANEL.MSG') reuse shr
ALLOC DSNAME('OPENFTQU.STD.CONN') DDNAME(OPENFT) SHR REUSE
TSOLIB ACT DATASET('OPENFTQU.OPENFT.NCLOAD')
```

2.2 Representation and utilization

You call the initial panel of the menu interface (Primary Option Menu, i.e. FTMAIN panel for openFT without FTAC or FTACM panel for openFT with FTAC) under TSO as follows:

```
EXECUTE '<openft qualifier>.OPENFT.PANEL.CLIST(FJMENU)'
```

The structure of the menu system is described below. Detailed help on each individual panel can be found online by pressing the F1 key.

Terminal operation is subject to the rules that usually apply with IBM ISPF:

- The ENTER key causes terminal input to be passed on and, where appropriate, verified. The particular reaction that follows depends on the panel currently displayed.
- In many cases, a data entry panel appears in which you can or must make entries. An action is then executed which has the same effect as issuing the corresponding FT command. In the next step, the message issued by openFT in response to this action is displayed on the screen. You exit this display in the usual manner using END, RETURN or the "jump function" (see below).
- The END command causes a return to the panel preceding the current panel in the panel hierarchy. In this case, no action is usually executed. This enables you to cancel actions which you have selected by mistake.
- The RETURN command causes a return to the Primary Option Menu. In this case, too, no action is executed.
- The "jump function" of ISPF (calling a sequence of panels in one step e.g. "=p.3") is supported. In this case, the effect of the END command differs from the one described above: as usual with ISPF, "the panel preceding the current panel in the panel hierarchy" is interpreted as the panel from which the "jump" was made; so the END command causes that panel to be displayed. In this case, too, no action is executed.
- Function keys PF1 through PF12 (or through PF24) can be used as usual with ISPF.
- If a syntax error or any other type of error is detected in the input, a short message is displayed in the top right-hand corner of the screen indicating the error. At the same time, the cursor is positioned at the input field concerned. Subsequent entry of the HELP command causes a more detailed message to be displayed in the third line on the screen. Repeating the HELP command causes a help panel to be displayed.
- The HELP panels for the individual functions form a hierarchy; you can therefore use the usual commands to "browse" through these help panels (e.g. ENTER to display the next help panel, BACK to display the previous help panel, etc.).
- Data you have entered in data entry panels is generally deleted as soon as you exit the panel. Exceptions are noted as appropriate for each panel.

- The data you have entered is not deleted, however, if the same data entry panel is displayed again following execution of the function (ENTER). This is the case for a number of functions which can be effectively repeated a number of times in succession (e.g. the function ADD REMOTE SYSTEM TO NETWORK DESCRIPTION). In this case, the data you have entered is also displayed once again and you can modify it before executing the function again. This applies until you finally exit the panel using END (or RETURN or the "jump function").
- In the case of "string" type input fields, the uppercase/lowercase notation is taken over, otherwise all inputs are converted to uppercase.
- The equals sign "=" has its usual ISPF navigation function (e.g. "=x" to exit the interface). For this reason, it is not possible to pass openFT any values that start with "=" via the interface.

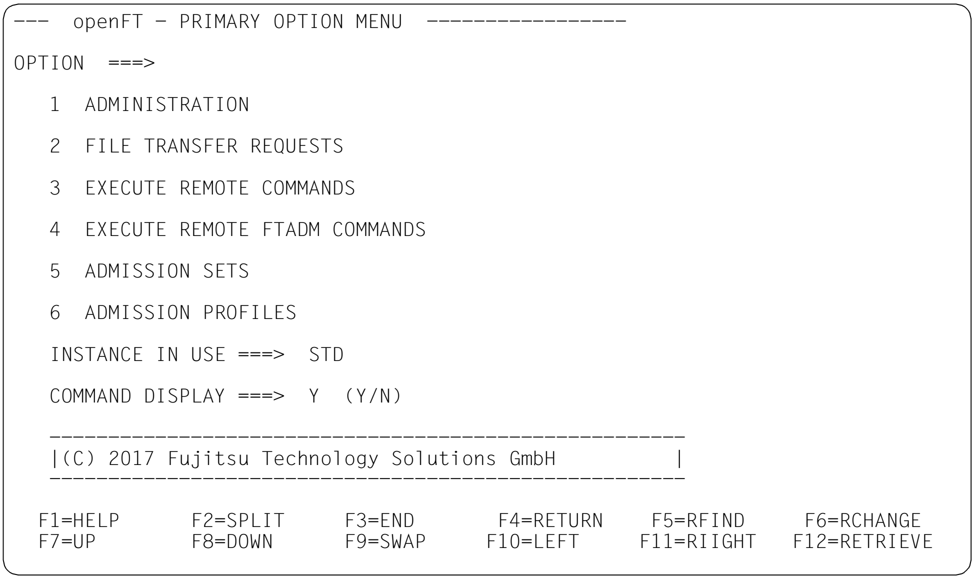
Refer to the relevant IBM manuals for further information about ISPF.

The entries you can or must make in the fields of the data entry panels correspond to the parameter values which you must specify for the corresponding FT administration command. They are described in the chapter “openFT commands” on page 97.

The messages issued by openFT in response to your actions are also the same as those issued at the command interface. These messages and their meanings are given in the manual "openFT (z/OS) - Installation and Operation".

openFT displays the "PRIMARY OPTION MENU" illustrated on the next page as the entry panel. Menu items 5 and 6 in this menu are only available if openFT-AC is installed.

PRIMARY OPTION MENU



```
---  openFT - PRIMARY OPTION MENU  -----------------

OPTION  ===>

   1  ADMINISTRATION

   2  FILE TRANSFER REQUESTS

   3  EXECUTE REMOTE COMMANDS

   4  EXECUTE REMOTE FTADM COMMANDS

   5  ADMISSION SETS

   6  ADMISSION PROFILES

   INSTANCE IN USE ===>  STD

   COMMAND DISPLAY ===>  Y  (Y/N)

   ______________________________________________________
   |(C) 2017 Fujitsu Technology Solutions GmbH          |
   ------------------------------------------------------

  F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIIGHT   F12=RETRIEVE
```

This is the initial panel of the menu interface for the FT administrator if FTAC is used. It is qualified as the "Primary Option Menu", which means that it is the panel to which you return from any subsequent panel after entering the RETURN command.

You enter YES or NO in the COMMAND DISPLAY field in order to specify whether or not the FT commands which correspond to the functions you select in the subsequent menus are to be displayed on the screen, together with all the parameters which correspond to your entries in the data entry panel, if applicable.

Provided you do not change this setting, it remains valid throughout the session and is retained after the session is terminated.

The following list illustrates the hierarchy of the subsequent menus and functions that can be accessed from the Primary Option Menu. FTAC-specific items are present only if openFT-AC is installed.

You can use the functions under 1 ADMINISTRATION only if you are the FT administrator.

1 ADMINISTRATION

- 1 OPERATING PARAMETERS
  - 1 LOAD openFT (ONLY AS A BATCH JOB)
  - 2 START LOCAL FT SYSTEM
  - 3 STOP LOCAL FT SYSTEM
  - 4 TERMINATE openFT
  - 5 KEY MANAGEMENT
    - 1 CREATE KEY SET
    - 2 DELETE KEY SET WITH REFERENCE ... (1..9999999)
    - 3 UPDATE KEY SET
    - 4 MODIFY KEY
    - 5 SHOW KEY
    - 6 IMPORT KEY
  - 6 MODIFY FT OPTIONS
- 2 REMOTE SYSTEMS

  (add, list, modify, remove FT-partners)
- 3 ADDITIONAL PARTNER DEFINITIONS (EDIT TNSTCPIP)
- 4 FTAC ENVIRONMENT
  - 1 EXPORT FTAC ENVIRONMENT
  - 2 IMPORT FTAC ENVIRONMENT
  - 3 SHOW FTAC ENVIRONMENT
- 5 INSTALLATION PARAMETERS
  - 1 DISPLAY DIALOG ENVIRONMENT
  - 2 DISPLAY PRESET INSTALLATION PARAMETERS
  - 3 EDIT PARM (INSTALLATION PARAMETERS)
  - 4 PRTJOB: ...

        - 5 TSOJOB: ...
        - 6 JCLJOB: ...
        - 7 TSOVVJOB: ...
        - 8 TSONVJOB: ...
        - 9 TSOVFJOB : ...
        - S SUCCMSG: ...
        - F FAILMSG: ...
    - 6 LOGGING/DIAGNOSTIC FUNCTIONS
        - 1 SHOW LOGGING RECORDS OR FILES
        - 2 DELETE LOGGING RECORDS OR FILES
        - 3 SHOW DIAGNOSTIC INFORMATION
        - 4 SHOW openFT TRACE DATA
- 2 FILE TRANSFER REQUESTS
    - 1 ENTER FILE TRANSFER REQUEST
    - 2 SHOW/MODIFY/CANCEL FILE TRANSFER REQUEST(S)
    - 3 SHOW LOGGING RECORDS OR FILES
    - 4 SHOW ALLOWED PARTNER SYSTEMS
    - 5 GET REMOTE FILES
- 3 EXECUTE REMOTE COMMANDS
- 4 EXECUTE REMOTE FTADM COMMANDS
- 5 ADMISSION SETS
- 6 ADMISSION PROFILES

  (Create, list, modify, delete FT admission profiles)

2.3 Error messages

The messages issued by openFT in response to your actions are the same as those issued at the command interface. These messages and their meaning are given in the manual "openFT (z/OS) - Installation and Operation".

Errors you make when entering data into the panels are displayed in the usual way in ISPF (output of a short message or, if the HELP command is issued, a long message).

Short messages and long messages can also occur for other reasons, however, e.g. in the event of errors when accessing temporary files. There are the following temporary files:

<inst>.FJCMD.TMP.OUT

When some of the menu interface functions are executed, a temporary PS data set is created to buffer the command. This data set is usually deleted again after the function has been executed.

inst: Instance name of the currently set openFT instance

<inst>.FJCMD.TMP.MSG

When some of the menu interface functions are executed, a temporary PS data set is created to buffer the messages generated by openFT. The content of this data set is automatically displayed on the screen (internal call of the PDF BROWSE service). Under normal circumstances, the data set is deleted when the display is closed.

inst: Instance name of the currently set openFT instance

If a temporary file cannot be created, you receive the following messages:

Short Message: `I/O - ERROR`

Long Message: `ERROR OCCURRED ON ACCESSING TEMPORARY OUTPUT FILE.`

2.4 Calling EDIT via the menu interface

As FT administrator, you can create and edit (EDIT) some members of the FT parameter library (e.g. PARM, TSOJOB, JCLJOB, PRTJOB, SUCCMSG, FAILMSG, TNSTCPIP) by means of the menu interface. The FT parameter library itself, however, must already exist.

When creating or modifying members, please note that they may contain **no line numbering**. You must therefore set NUMBER OFF in your EDIT profile.

If you call the PDF service EDIT via the menu interface for the FT administrator, the "recovery" function is also available. This means that:

– you can use the UNDO command.

– Following a system failure during an EDIT session, the old session is first recovered the next time you call EDIT (for any member). This is indicated by the following messages:

  Short Message: `CAUTION – RECOVERY`

  Long Message: `THIS IS EDIT RECOVERY OF MEMBER...`

  You can now continue with this EDIT session. If you exit this session, the member you originally selected is edited.

  Following a system failure, you must call PDF-EDIT in the same way as before, i.e. either via the menu interface for the FT administrator or via the general ISPF/PDF interface of your system.

  Otherwise it is not possible to recover the session which was interrupted by the system failure.

2.4.1 Error messages for EDIT

If errors occur when the PDF service EDIT are called via the menu interface for the FT administrator, an "ISPF DIALOG ERROR" screen is displayed. The error messages displayed here have the following meaning:

```
DATA SET NOT CATALOGED
'.....' WAS NOT FOUND IN CATALOG.
```

Meaning
No FT parameter library exists or the file is not catalogued.

Response
Create or catalog the FT parameter library (PO or PDSE data set), observing the notes provided in the section „Setting up the FT parameter library“ in manual "openFT (z/OS) - Installation and Operation".

```
AUTHORIZATION FAILED
YOU MAY NOT USE THIS PROTECTED DATA SET. OPEN 913 ABEND.
```

Meaning
You may not access the (RACF-protected) FT parameter library.

Response
Modify (or ask the administrator to modify) the access rights for the FT parameter library.

```
MEMBER IN USE
MEMBER IS BEING UPDATED BY YOU OR ANOTHER USER.
```

Meaning
The selected member of the FT parameter library is already being processed by another FT administrator.

Response
Coordinate modifications to members of the FT parameter library.

3 Introduction to the command interface

This chapter contains a functional description of the openFT commands for users and administrators.

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 chapter “openFT commands” on page 97.

All commands described in the manual can be entered directly at a TSO dialog terminal.

i For information on how the FT administration commands can be used in a z/OS system without TSO, consult the manual "openFT (z/OS) - Installation and Operation"

The commands FJGEN, FJGENPAR, FJUPDPAR and FJINIT are described in the manual "openFT (z/OS) - Installation and Operation".

Requirements

In order to use the menu interface, the following requirements have to be fulfilled:

- The libraries necessary for using the command interface are concatenated for the desired instance.

For details, see the manual "openFT (z/OS) - Installation and Operation".

Setting an openFT instance

The commands always apply to the currently set openFT instance. If you want to use an instance other than STD you must set the instance:

– The CLIST <openft qualifier>.<inst>.CLIST must be concatenated

– The file <openft qualifier>.<inst>.CONN must be allocated with the DD name OPENFT.

The FT administrator defines the specifications for the OPENFT QUALIFIER and the name of the instance (<inst>) when setting up the instance.

In a running TSO session, you can perform these allocations manually. This means, for example, that you can also change the openFT instance within the TSO session. You should store the necessary commands in a CLIST and execute these in TSO mode. You can also record the name of this CLIST in a LOGON procedure so that the commands are always executed when you log on.

Example of this type of CLIST with instance INST2

```
ALLOCATE FILE(SYSPROC) DATASET('ISP.SISPCLIB') REUSE SHR
ALLOC DSNAME('OPENFTQU.INST2.CONN') DDNAME(OPENFT) SHR REUSE
TSOLIB ACT DATASET(’OPENFTQU.OPENFT.NCLOAD’)
```

3.1 Functional command overview

The following overview shows the FT and FTAC 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 sets and admission profiles 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.

3.1.1 FT commands overview

Transfer files and manage request queue		
Submit asynchronous FT request	NCOPY	page 350
Submit synchronous FT request	FTSCOPY	page 255
Fetch multiple files from a remote system	FTMGET	page 169
Cancel FT requests	NCANCEL, also FTCANREQ	page 343
Show information on FT requests	NSTATUS, also FTSHWREQ	page 394
Modify FT request queue	FTMODREQ	page 251
File management		
Show attributes of file/files in a remote system	FTSHW	page 260

Change file attributes in a remote system	FTMOD	page 179
Delete file in a remote system	FTDEL	page 139
Create directory in a remote system	FTCREDIR	page 114
Change attributes of a directory in a remote system	FTMODDIR	page 191
Delete directory in a remote system	FTDELDIR	page 142
Execute remote commands		
Execute commands in the remote system	FTEXEC	page 154
Administer openFT partners		
Add partner to the partner list	FTADDPTN	page 98
Remove partner from the partner list	FTREMPTN	page 254
Modify partner properties	FTMODPTN	page 243
Display partner systems	FTSHWPTN	page 327
Activate and deactivate or terminate openFT		
Activate openFT	FTSTART	page 338
Deactivate openFT	FTSTOP	page 339
Terminate openFT	FTTERM	page 340
Controlling openFT operating parameters		
Modify operating parameters	FTMODOPT	page 196
Display operating parameters	FTSHWOPT	page 313
Administer key pair sets for authentication		
Create a key pair set	FTCREKEY	page 117
Import keys	FTIMPKEY	page 166
Display key properties	FTSHWKEY	page 278
Update public keys	FTUPDKEY	page 341
Modify keys	FTMODKEY	page 194
Delete a key pair set	FTDELKEY	page 145

Showing openFT instances

Show openFT instance	FTSHWINS	page 277

Logging Function

Delete log records or offline log files	FTDELLOG	page 146
Show log records or log files	FTSHWLOG	page 281
Display information on reason codes in the logging records	FTHELP	page 161

Monitoring

Show monitoring data	FTSHWMON	page 301

Obtain information on openFT

Display the network environment	FTSHWNET	page 312
Display operating parameters	FTSHWOPT	page 313

Remote administration

Issue remote administration command	FTADM	page 106

3.1.2 FTAC commands overview

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

Edit FTAC admission profiles		
Create admission profile	FTCREPRF	page 119
Delete admission profile	FTDELPRF	page 151
Modify admission profile	FTMODPRF	page 220
Show admission profile	FTSHWPRF	page 321
Edit FTAC admission sets		
Modify admission set	FTMODADS	page 184
Show admission set	FTSHWADS	page 270
Store and display saved FTAC admission profiles and sets		
Export admission profiles and sets	FTEXPENV	page 159
Import admission profiles and sets	FTIMPENV	page 163
Display saved admission profiles and sets	FTSHWENV	page 274
Show partner systems		
Display partner systems and security levels	FTSHWRGE	page 335

3.2 Entering FT commands

Please remember the following when entering commands:

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

  ```
  NCOPY TRANSFER-DIRECTION=TO,PARTNER-NAME=ZENTRALE,LOCAL-PARAMETER =...
  ```

- If quotes appear in a value assignment which is itself enclosed in quotes, they must be entered twice.
- If there is no default value marked (by underscoring) for an operand, then it **must** be specified with a valid value (mandatory operand).
- A distinction is made between positional operands and keyword operands. Positional operands are uniquely determined by their position in the command. Keyword operands are uniquely determined by their keyword, for example `TRANSFER-DIRECTION=...` There are a number of considerations to be borne in mind when specifying such operands (see below).
- You can abbreviate your entries for operands, always ensuring that your entries retain their uniqueness. You can also use positional operands if you wish. Short forms and long forms can be mixed at will. Certain abbreviated forms of keywords and a number of positional operands are guaranteed for openFT. In the command representation the recommended abbreviation is shown in **bold**. This means that you will find these options unchanged in subsequent versions. This means, therefore, that to be “on the safe side”, you should form the habit of entering these commands in their abbreviated form. You should take particular care to use the guaranteed abbreviated forms in procedures, as this will ensure their continued executability in subsequent versions. The recommended abbreviations are used in the examples shown in this chapter. The possible abbreviations are listed for the individual command formats.
- If a structure is preceded by an introductory operand value, then the opening parentheses must immediately follow this operand value. Example: *BS2000 is an introductory operand value in REM=*BS2000(...). Introductory operand values may be omitted if there is no risk of ambiguity.
- The asterisk (*) that precedes constant operand values may be omitted if there is no risk of ambiguity. Please ensure that it is not a guaranteed abbreviation.
- Comments may be included in FT user commands using the form "..."; the normal method of including comments in other TSO commands using the form /*...*/ is not permitted.
- Please note that at the command interface, a c string must also be entered in the form C’...’ as otherwise (without the C) openFT would try to interpret the string as a fully qualified z/OS file name.

When you enter commands, the value assignments for the operands may be specified in positional form, in keyword form or in mixed form.
Please note the following:

- When you perform value assignments in positional form, the first value is assigned to the first operand in the command, the second value to the second operand etc.
- Values assigned in positional form are separated by commas. You must also enter a comma for each operand for which no value is assigned.
- If two values are assigned to an operand, the last value to be assigned always applies. This also applies to parameter specifications in introductory operand values within the corresponding structure brackets. However, for the sake of clarity, double assignments should generally be avoided.
- If you mix the different forms of operand value assignments (positional and keyword form), then you must observe the correct sequence. Note that you can start your input with positional operands and follow these with keyword operands but not the other way round!
- Since there is a possibility that the sequence of operands may change in subsequent versions, only keyword operands should be used in procedures.

Continuation lines in FT commands in z/OS

An NCOPY command may consist of more than one line. When entering an NCOPY command with continuation lines at a TSO terminal, you simply continue writing on the next line on the screen.
If an NCOPY command with continuation lines is issued in a CLIST or REXX procedure or in a batch job as data for the IBM utility IKJEFT01, a hyphen "-" or a plus sign "+" is used as the continuation character. Refer to the IBM manuals for more details.

Differentiation between uppercase and lowercase letters

It may be important to differentiate between uppercase and lowercase letters in the parameters.

openFT handles the letters contained in the command string according to the following rules:

1. If the command string received by openFT contains only uppercase letters,
   - all letters outside the quotation marks remain uppercase;
   - letters enclosed in quotation marks are converted to lowercase.
   - alphanumerically specified FTAC transfer admissions are converted into lowercase letters

2. If any part of the command string received by openFT except the command name (NCOPY) contains a lowercase letter,
   - all letters outside the quotation marks are converted to uppercase;
   - alphanumerically specified FTAC transfer admissions are converted into lowercase letters
   - letters enclosed in quotation marks are not converted. These letters are retained in the form in which they were entered.

This has the following consequences for command input:

If parameter values consisting of uppercase letters (or of both uppercase and lowercase letters) enclosed in quotation marks are to be entered, you must ensure that

- the command contains at least one lowercase letter (at any position except in the command name) and
- openFT receives this command string in the same form (with no conversion).

This means that

- In a CLIST or REXX procedure, you must use the statement CONTROL ASIS (or CONTROL NOCAPS) to ensure that the command string is not converted to uppercase before execution.
- You can also use the menu interface (see page 21); here, the relevant fields are not converted to uppercase (see the description of the input fields in the data entry panels).
- When the TSO command processor is called in a batch job (IBM utility IKJEFT01, see manual "openFT (z/OS) - Installation and Operation"), letters are not converted to uppercase.

These rules also apply to the hexadecimal digits A through F in entries of the form <x-string m..n> which expect the partner system to be specified in uppercase letters.

3.3 Command syntax representation

The command format consists of a field with the command name. All operands with their legal values are then listed. Operand values which introduce structures and the operands dependent on these operands are listed separately. The syntax of the command representation is explained in the following three tables.

Table 1: Notational conventions

The meanings of the special characters and the notation used to describe command and statement formats are explained in table 1.

Table 2: Data types

Variable operand values are represented in SDF by data types. Each data type represents a specific set of values. The number of data types is limited to those described in table 2.

The description of the data types is valid for the entire set of commands/statements. Therefore only deviations (if any) from the attributes described here are explained in the relevant operand descriptions.

Table 3: Suffixes for data types

Data type suffixes define additional rules for data type input. They contain a length or interval specification.

The description of the data type suffixes is valid for the entire set of commands/statements. Therefore only deviations (if any) from table 3 are explained in the relevant operand descriptions.

Metasyntax

Representation	Meaning	Examples
UPPERCASE LETTERS	Uppercase letters denote keywords (command, statement or operand names, keyword values) and constant operand values. Keyword values begin with *	**HELP-SDF** **SCREEN-STEPS** = ***NO**
UPPERCASE **LETTERS** in boldface	Uppercase letters printed in boldface denote guaranteed or suggested abbreviations of keywords.	**GUID**ANCE-**MODE** = ***Y**ES
=	The equals sign connects an operand name with the associated operand values.	**GUID**ANCE-**MODE** = ***NO**
< >	Angle brackets denote variables whose range of values is described by data types and suffixes (see Tables 2 and 3).	**SYNTAX-F**ILE = <filename 1..54>
Underscoring	Underscoring denotes the default value of an operand.	**GUID**ANCE-**MODE** = ***NO**
/	A slash serves to separate alternative operand values.	**NEXT-FIELD** = ***NO** / ***Y**ES
(…)	Parentheses denote operand values that initiate a structure.	,**UNGUID**ED-**DIA**LOG = ***Y**ES(...) / ***NO**
[ ]	Square brackets denote operand values which introduce a structure and are optional. The subsequent structure can be specified without the initiating operand value.	**SELECT** = [***BY-ATTR**IBUTES](...)
Indentation	Indentation indicates that the operand is dependent on a higher-ranking operand.	,**GUID**ED-**DIA**LOG = ***Y**ES(...) ***Y**ES(...) **SCREEN-STEPS** = ***NO** / ***Y**ES

Table 1: Metasyntax (part 1 of 2)

Representation	Meaning	Examples
│	A vertical bar identifies related operands within a structure. Its length marks the beginning and end of a structure. A structure may contain further structures. The number of vertical bars preceding an operand corresponds to the depth of the structure.	**SUP**PORT = ***TAPE**(...) ***TAPE(...)** │ **VOL**UME = ***ANY**(...) │ ***ANY**(...) │ │ ... │ │
,	A comma precedes further operands at the same structure level.	**GUID**ANCE**-MODE** = ***NO** / ***Y**ES ,**SDF-COM**MANDS = ***NO** / ***Y**ES
list-poss(n):	The entry “list-poss” signifies that a list of operand values can be given at this point. If (n) is present, it means that the list must not have more than n elements. A list of more than one element must be enclosed in parentheses.	list-poss: ***SAM** / ***ISAM** list-poss(40): <structured-name 1..30> list-poss(256): ***OMF** / ***SYSLST**(...) / <filename 1..54>
Alias:	The name that follows represents a guaranteed alias (abbreviation) for the command or statement name.	**HELP-SDF** Alias: **HPSDF**

Table 1: Metasyntax (part 2 of 2)

Data types

Data type	Character set	Special rules
alphanum-name	A…Z 0…9 $, #, @	
c-string	EBCDIC character	Must be enclosed within single quotes; the letter C may be prefixed; in the case of file names in z/OS it must be prefixed; any single quotes occurring within the string must be entered twice.
composed-name	A…Z 0…9 $, #, @ Hyphen Period	Alphanumerical string that can be subdivided into multiple substrings by periods or hyphens.
date	0…9 Structure identifier: hyphen	Input format: yyyy-mm-dd yyyy: year; optionally 2 or 4 digits mm: month dd: day Only date specifications between 1.1.2000 and 19.1.2038 are possible. If the year is specified in 2-digit form, 2000 is added to the number
filename	A…Z 0…9 $, #, @ hyphen period Colon Single quote	Input format fully qualified: ':<prefix>:<first-qual>.<filename>' Input format partially qualified: :<prefix>:<filename> :<prefix>: Optional specification of file organization; enclosed in colons; can assume the following values: :S: for PS :O: for PO :E: for PDSE :L: for PO or PDSE :V: for VSAM

Table 2: Data types (part 1 of 3)

Data type	Character set	Special rules
		<first-qual> "first level qualifier" User ID (max. 7 characters, character range A…Z, 0…9, $, #, @; may not begin with a digit) or alias (max. 8 characters) <filename> partially qualified file name; the syntax of z/OS file names depends on the file organization; refer to the overview section "Transferring z/OS files" on page 50
filename-prefix	A…Z 0…9 $, #, @ hyphen period Colon Single quote	Input format fully qualified: ':<prefix>:<first-qual>.<partname>.' or ':<prefix>:<first-qual>.<partname>/' Input format partially qualified: :<prefix>:<partname>. or :<prefix>:<partname>/ <prefix> see filename <first-qual> see filename partname Specifies the common first part of the partially qualified name of files. partname must be followed by a period or a slash.
integer	0…9, +, -	+ or -, if specified, must be the first character.
name	A…Z 0…9 $, #, @	Must not begin with 0...9.
number	0...9 A...F	Message number/return code

Table 2: Data types (part 2 of 3)

Data type	**Character set**	**Special rules**
partial-filename	A…Z 0…9 $, #, @ hyphen period	Input format fully qualified: ':<prefix>:<first-qual>.<partname>.' Input format partially qualified: :<prefix>:<partname>. <prefix> see filename <first-qual> see filename partname Specifies the common first part of the partially qualified name of files. partname must be followed by a period.
text	freely selectable	For the input format, see the relevant operand descriptions.
time	0…9 structure identifier: colon	Time-of-day entry: Input format: $\left\{\begin{array}{l}\text{hh:mm:ss}\\ \text{hh:mm}\\ \text{hh}\end{array}\right\}$ $\left.\begin{array}{ll}\text{hh:} & \text{hours}\\ \text{mm:} & \text{minutes}\\ \text{ss:} & \text{seconds}\end{array}\right\}$ Leading zeros may be omitted Valid entries are between 00:00:00 and 23:59:59.
x-string	Hexadecimal: 00…FF	Must be enclosed in single quotes; must be prefixed by the letter X. There may be an odd number of characters.

Table 2: Data types (part 3 of 3)

Suffixes for data types

Suffix	Meaning
x..y	With data type "integer": interval specification x minimum value permitted for "integer". x is an (optionally signed) integer. y maximum value permitted for "integer". y is an (optionally signed) integer.
x..y	With the other data types: length specification For data types date and time the length specification is not displayed. x minimum length for the operand value; x is an integer. y maximum length for the operand value; y is an integer. x=y the length of the operand value must be precisely x.

Table 3: Suffixes for data types

Meaning of operands

After the format of each command there is a detailed description of all the operands, the possible value assignments and their functions.

Otherwise the same metasyntax is used in describing operands as in the representation of the command formats (see above).

The following characters are regarded as constants in describing the operands: "." (period), "(" (open bracket), ")" (close bracket), " ' " (single quote), "$" (dollar sign), and also the character combinations ":V:", ":L:", ":S:", ":O:" and ":E:" i.e. they must be specified when the command is entered. Where this occurs the syntactical components of the operand value must follow one after another without any gaps.
"±" has the usual meaning "+" or "-".

Example

Possible entries for the local operand FILE are as follows:

```
ABC                  'USER1.ABC',                    (1)
GROUP1.G1234V01      'USER1.GROUP1.G1234V01'         (2)
GROUP2(+27)          'USER1.GROUP2(+27)'             (3)
GROUP3(0)            'USER1.GROUP3(0)'               (4)
:V:VSDAT             ':V:USER1.VSDAT'                (5)
PDS1(DEF)            'USER1.PDS1(DEF)',              (6)
:L:PODS2             ':L:USER1.PODS2'                (7)
./directory5/abcd    /u/user002/directory5/abcd      (8)
```

Key

(1) Name of a PS data set

(2) Name of an absolute generation data set (PS data set) (this has the same syntax as the name of a normal PS data set, with the exception of the last partial name, which must have a special format)

(3) Name of a relative generation data set (PS data set)

(4) Name of a relative generation data set (PS data set), special case "current generation" (may only be a send file)

(5) Name of a VSAM file of the type "entry sequenced"

(6) Name of a PO or PDSE member

(7) Name of an entire PO or PDSE data set

(8) Pathname of an openEdition file (absolute and relative)

More details on the syntax rules for file names, passwords, user IDs and account numbers in openFT can be found in the respective sections in chapter 3.

3.4 Transferring z/OS files

openFT (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.

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. With BS2000 systems, for example, SAM files and PLAM elements of the appropriate type can be exchanged.

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

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

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

Primary and secondary allocation

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

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

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

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

Encoding

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

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

The following files cannot be transferred by openFT:

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

3.4.1 Files that can be transferred

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

Local openFT (z/OS)	**Remote FT system**	**Type of file**
PO data set PDSE data set	openFT (z/OS)	PO data set PDSE data set
PS data set PO member PDSE member VSAM-ES data set openEdition file	openFT (z/OS)	PS data set PO member PDSE member VSAM-ES data set openEdition file
	openFT (BS2000)	SAM file LMS element POSIX file
	other FT system	sequential file

3.4.2 Volumes

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

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

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

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

3.4.3 Attributes of receive files

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

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

The following points also apply:

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

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

The Data Management System also issues an error message.

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

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

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

Type of receive file	**Existing file with the same name**	**Value of the WRITE-MODE operand**		
		*NEW	*REPLACE	*EXTEND
PS data set	no	N	N	N
	yes	---	O	E
VSAM file	no	N	N	N
	yes	---	D + N	E
openEdition file	no	N	N	N
	yes	---	D + N	E

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

Explanation:

N A new receive file is created.

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

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

E The file is extended.

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

For a better understanding of the processes involved in member transfer, refer to section “Notes on transferring entire PO or PDSE data sets” on page 69.

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

Most of the attributes of a receive file, however, are determined by the corresponding attributes of the send file. The following factors determine the extent to which the file attributes of these two files can be mapped onto one another:

- Which attributes are transferred via the link involved?
  A distinction is made between the following types of link:
  - **Homogeneous** link: Link between two openFT (z/OS) systems
  - **Heterogeneous** link: Link between openFT (z/OS) system and another FT system, e.g. openFT (BS2000)

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

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

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

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

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

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

3.4.3.1 Attributes of new receive files of any type

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

File structure:

The file structure of the receive file is derived from the prefix specified as part of the file name in the NCOPY command. The following prefixes can be used (see also page 54):

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

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

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

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

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

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

:L:filename represents a complete PO or PDSE data set; the FT administrator can specify the file structure PO or PDSE; if the FT administrator has not entered a predefined value, the following default values are used:

- if the IBM software product DFSMS (Data Facility System-Managed Storage) is not installed, PO is the default
- if DFSMS is active: the default is specified with an ACS routine or SYS1.PARMLIB (please consult your z/OS system administrator)

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

File name:

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

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

Volume:

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

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

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

Storage allocation:

For both homogeneous and heterogeneous links, the allocation of disk storage to the receive file is determined by the transferred file size of the send file (see “Primary and secondary allocation” on page 50).

Exception

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

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

Data access control:

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

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

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

3.4.3.2 Attributes of new receive files of specific types

This section describes the attributes of newly created receive files of the following types:

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

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

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

Record format:

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

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

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

VSAM file

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

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

U :
cannot be mapped → FTR2096

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

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

Notes

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

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

  DATA-TYPE=*BIN → record format attribute U

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

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

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

Block length:

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

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

The following applies to the different types of receive file:

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

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

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

Record length:

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

F(ixed) Every record has this length.

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

> \> 0 : The length of a record may not exceed this length.
>
> = 0 : A record can have any length, with 32752 byte as the maximum (without record length field). However, there are restrictions depending on the file type, see below.

U(ndefined) No meaning (= 0)

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

The following applies to the different types of receive files:

PS data set:

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

record length transferred = 0:

U format : OK.

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

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

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

PO/PDSE member

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

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

record length transferred = 0:

U format : OK.

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

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

VSAM file:

record length transferred > 0:

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

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

record length transferred = 0:

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

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

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

Notes

- The maximum length of the records that are to be transferred may not exceed the following values:
  - 32760 byte for files with fixed record lengths
  - 32752 byte for files with variable record lengths (record length without record length field)
  - for openFT ≤ V10.0: 32248 byte with compromised transfer (COMPRESS = *BYTE-REPETITION)

- So it can be assumed that, send files from remote systems which are to be transferred in format V with a record length of = 0 cannot be stored in newly created receive files under the following circumstances.

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

  In either case, the transfer is canceled with an error message.

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

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

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

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

Spanned records attribute:

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

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

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

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

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

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

Blocked records attribute:

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

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

Note

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

Control character attributes:

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

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

Number of directory blocks:

PS data set, VSAM file: Not applicable

PO/PDSE member

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

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

Number of directory blocks:

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

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

3.4.4 Transferring a PO or PDSE member

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

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

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

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

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

These specifications have the following meaning:

*REPLACE-FILE: The member is written to the data set even if a member with the same name already exists.
For a PO data set applies: The new member is then located at the end of the PO data set; there may be a gap in place of the old member.

*NEW-FILE: Transfer takes place only if no member with the same name exists.

*EXTEND-FILE: The contents of any existing member with the same name are copied. The transferred data is appended and the resulting member is written to the data set.
For a PO data set applies: The extended member is then located at the end of the PO data set; there may be a gap in place of the old member.

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

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

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

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

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

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

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

3.4.5 Transferring a PO or PDSE data set

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

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

- for any data set (PO or PDSE):

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

- for access to a specific PO data set:

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

- for access to a specific PDSE data set:

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

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

*REPLACE-FILE: If a data set with the same name already exists, it is overwritten in its entirety.

*NEW-FILE: Transfer takes place only if no data set of the same name already exists.

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

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

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

– The receive file identified as a PO or PDSE data set is generated from this temporary PS data set by means of IEBCOPY-Load in the remote system. You can specify the type of the data set (PO or PDSE) using the prefixes mentioned above (":O:" for PO, ":E:" for PDSE). If you do not specify a prefix, or you specify the prefix ":L:", the dataset type of the send file (PO or PDSE) is used as the type of the newly created dataset if openFT as of V10 is running on the z/OS partner system, otherwise the default value specified by the FT administrator (prefix ":L:") is used.

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

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

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

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

transuid
: is the user ID which was specified in the TRANSFER-ADMISSION for the system involved. This also applies if the name of the PO or PDSE data set was specified in its fully qualified form in the NCOPY command. In this case, the "first level qualifier" for the temporary PS data set is replaced by this user ID.

pdsname
: Partially qualified name of the PO or PDSE data set.

.U
: This suffix identifies the temporary PS data set. For this reason, the length of the PO or PDSE data set itself may not exceed 42 characters (fully qualified) or 34 characters (partially qualified).

Volume:
: openFT creates the temporary PS files on a volume specified by the FT administrator (the same volume is used for all transfer requests). If the FT administrator has not specified a volume, the default volume defined for new files on the system involved applies.

Storage requirements:
: About the same as for the associated PO or PDSE data set

Other attributes:
: Default values generated by IEBCOPY, i.e.: RECFORM= VS, BLKSIZE= max(284, PO-DS-BLKSZIZE + PO-KEYLEN + 20), RECSIZE= BLKSIZE - 4

Please refer to the note on data protection on page 70.

3.4.5.1 Notes on transferring entire PO or PDSE data sets

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

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

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

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

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

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

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

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

  In this case, the transfer itself is already complete.

3.4.5.2 Note on data protection

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

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

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

Example

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

'MILLER.*.U'

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

3.4.6 Transferring a generation data set

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

openFT cannot transfer entire generation data groups.

Absolute generation data set

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

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

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

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

nn is the version number (between 00 and 99)

Relative generation data set

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

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

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

(...) is the relative generation number:

(0) current generation

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

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

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

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

Generation number

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

relative generation number (0):

absolute generation number = current generation number

relative generation number (-n):

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

relative generation number (+n):

absolute generation number = current generation number + n modulo 9999

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

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

Version number

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

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

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

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

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

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

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

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

Causes of the error message FTR0020

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

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

This error message may be caused by:

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

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

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

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

- If the absolute generation number G9999 is exceeded, modulo 9999 applies (see above), i.e. the generation number reverts to zero in the sequence of counting. This can have unexpected results if an absolute generation number calculated in this way has already been assigned to existing generations.

- When a new generation is created, the oldest generation or all previous generations may be deleted (depending on the LIMIT, EMPTY/NOEMPTY and SCRATCH/NOSCRATCH parameters specified in the DEFINE-GENERATIONDATAGROUP command when the generation data group was created).

- For files in the local system, the relative generation data set name is converted to an absolute generation data set name when the request is accepted. For files in the remote system, this conversion takes place at the start of request processing. Once a generation name has been assigned, it remains the same throughout request processing. This has the following results:
  - If, once the absolute generation data set name has been formed, new file generations are created (e.g. by a batch job executing in parallel), the resulting displacement of the "current" generation data set number is not taken into account.
  - The relative generation data set name appears in the result list and in the transfer logging function (FT administrator function, see the manual "openFT (z/OS) - Installation and Operation") exactly as it was specified in the NCOPY command.
  - Similarly, in a NSTATUS or NCANCEL command, you must specify the relative generation data set name exactly as it was specified in the original NCOPY command.
- There is no way of ensuring that two transfer requests cannot access the same generation data group. If this occurs, the generation numbering may be displaced unexpectedly.
- In an FT system which does not support the specification of relative generation data set names for REM=*MSP, the REM=*ANY syntax can be used to access a relative generation data set as a send or receive file in a remote system of type openFT (z/OS).
- Inbound file management requests cannot access relative file generations.

3.5 Command return codes

The TSO commands supply a return code that provides information about whether command processing has succeeded or failed. It is stored in the TSO's system variable ("control variable") &LASTCC. A return code other than 0 is generated only if a corresponding message is output at the terminal. These messages are described in the manual "openFT (z/OS) - Installation and Operation".

This return code may have the following values:

`Return-Code = 0`:

The command was accepted. (Corresponds, for example, to the message FTR0000 or FTR0008 at the terminal.)

`Return-Code = 4`:

The command was accepted with a minor warning, for example if no corresponding administration objects were found.

`Return-Code = 8`:

Reserved

`Return-Code = 12 (or > 12):`

The command was rejected due to an error. The request was not accepted.

The TSO commands can also be started in response to an ftexec command that was started in a remote Unix or Windows partner system. The partner system is sent either the return code 0 (if the command was accepted) or 12 (if the command was terminated with an error).

3.6 Specifying partner addresses

The following applies to the addressing of a partner system:

- You can specify the name of the partner from the partner list provided that the partner has been entered in the partner list.

  A partner has to be entered in the partner list by the FT administrator. For this purpose, the FT administrator can use the following commands:

  – FTADDPTN

  – FTMODPTN

- You can access a partner directly via its address in FT or file management requests even if it is not entered in the partner list. This is only possible if the “dynamic partner” function is enabled via operating parameters.

Partner addresses

A partner address has the following structure:

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

host (= computer name, see page 76) 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 78. Final ’.’ or ’:’ can be omitted.

The individual components of the address have the following meanings:

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

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

Default value: **openft**

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

- internet host name (e.g. DNS name), length 1 to 80 characters, up to 24 characters for z/OS partner systems with openFT < V12.1.
- TNS name from the z/OS library (TNSTCPIP member), up to 8 characters in length.
- SNA LU name, length 1 to 8 characters.
- IPv4 address with the prefix %ip, i.e. for example %ip139.22.33.44.
  The IP address must always be specified as a sequence of decimal numbers separated by dots and without leading zeros.

port

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

Default value:	**1100** for openFT partners. A different default value can also be set in the operating parameters using the following command: FTMODOPT
	21 for FTP partners
	11000 for ADM partners

tsel

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

- 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**

Printable transport selectors are always used in uppercase in openFT even if they are specified or output in lowercase.

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 is encoded in ASCII with a variable length in the log.

Default value: empty

Printable session selectors are always used in uppercase in openFT even if they are specified or output in lowercase.

psel

Not used under z/OS.

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
FTP partner	ftp://FILESERV
FTADM partner	ftadm://FILESERV
SNA partner via openFT protocol (FILESERV is the LU name)	FILESERV:sna

3.7 Entering the authorization data for the partner system

The authorization data can be specified via login/LOGON authorization or via FTAC transfer admission, see the following table:

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 (Unicode characters are also permitted) 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 (Unicode characters are also permitted)	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 LOGON/login authorization for the individual platforms using the following syntax:

- BS2000 systems:

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

  You can omit the account number if the user has a default account number for the BS2000 timesharing mode and you want to use this default account number.

- Unix systems

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

- Windows systems:

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

The user ID consists of a user name (In the case of local IDs, the "`hostname\`" 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 a backslash (\) and the user name.

- OS/390 and z/OS:

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

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

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

Inbound access using the default FTP client

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

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

  ```
  ftp hostname
  ```

  hostname is the host name of the openFT FTP server.

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

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

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

  Example

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

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

  Example

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

3.8 Preprocessing and postprocessing

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

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

In the case of preprocessing only the command’s SYSPRINT or %TEMPFILE output is transferred. The SYSPRINT or %TEMPFILE output is temporarily stored in a file which is deleted following transfer. This file is created with a unique file name in order to prevent conflicts between file processing operations that are running in parallel. The ID under which preprocessing is running must possess sufficient space for the creation of the temporary file as otherwise preprocessing will be aborted.

The temporary files that are created for pre/postprocessing are all of type "Variable Blocksize (VB)", and are automatically deleted as soon as transfer and/or preprocessing and postprocessing are completed.

Preprocessing and postprocessing jobs generated by openFT

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

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

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

  Examples

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

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

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

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

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

3.9 Follow-up processing

***DELETE as a special form of follow-up processing**

This character string *DELETE 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.

3.9.1 Syntax rules for follow-up processing on z/OS

The character string enclosed in single quotes may consist of:

- one or more TSO commands separated from one another by semicolons:
  'command' or 'command;command;command...'
- one or more JCL statements separated from one another by semicolons: 'statement' or
  'statement;statement;statement...'
  You can specify the following JCL statements which are identified by their JCL identifier:
  // "real" JCL statements
  //* comment lines
  no JCL identifier: in-stream data
- the specific openFT statement 'ALLOC DSNAME(filename)', if a user-created PS data set containing all JCL statements for follow-up processing is to be allocated or
- the specific openFT statement 'ALLOC DSNAME (filename(member-name))' if a user-created PO or PDSE data set is to be allocated whose member "member-name" contains all JCL statements for follow-up processing.

The specific openFT statement 'ALLOC DSNAME(...' must be entered exactly as specified above. If the TSO command ALLOC is to be specified as follow-up processing, then - unlike the special openFT statement ALLOC DSNAME - the full TSO command name must be specified (i.e. ALLOCATE DSNAME ...).

If one or more TSO commands or one or more JCL statements are specified as follow-up processing, then openFT generates a job that is passed to the system via the internal reader. The FT administrator can adapt uniform job envelopes that are based on the follow-up processing commands surrounding it to the circumstances of the local system.

3.9.2 Variables for follow-up processing

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 section “Specifying partner addresses” on page 76. In addition, all characters apart from letters, digits or periods are replaced by ’@’.

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.

Additional variables on z/OS

The following variables can be used:

`%FILN` Like %FILENAME.

`%FILX` Like FILN, except that quotes are automatically doubled during the replacement.

`%PNAM` Symbolic name of the partner system. A maximum of 8 characters.

%USID User ID from the PROCESSING-ADMISSION for the relevant system, if specified, otherwise from the TRANSFER-ADMISSION. A maximum of 7 characters in accordance with IBM conventions.

%ACCN Accounting information from the PROCESSING-ADMISSION for the relevant system, if specified, otherwise from the TRANSFER-ADMISSION. A maximum of 40 characters in accordance with IBM conventions.

%ACCX Like ACCN, except that quotes are automatically doubled during the replacement.

%PASS Password from the PROCESSING-ADMISSION for the relevant system, if specified, otherwise from the TRANSFER-ADMISSION. A maximum of 8 characters in accordance with IBM conventions.

%OWID Owner of the FT job, i.e. the user ID under which the transfer job was created. A maximum of 7 characters in accordance with IBM conventions. This parameter is replaced only in the system in which the transfer job was created. It is eliminated in the partner system. It should therefore be specified only in follow-up processing for the local system.

%PGRN Programmer's name, as specified with the keyword PGRN= as a subcommand in the command string for follow-up processing in the relevant system (see section “Specifying a programmer's name” on page 85). A maximum of 20 characters in accordance with IBM conventions.

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

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

For reasons of compatibility with previous versions, the character "&" should be used to make sure that previous versions also handle the variables correctly.

The names of these variables should have as many trailing "#" fill characters as are necessary for a field to be set to its maximum length (including the "&" character, e.g. &TRID######). When replacing the variables by the current values, openFT does not exceed the field length predefined by the name of the symbolic parameter including the trailing "#" fill characters; if necessary the current values are truncated. On the other hand, where the current values are shorter than this field length, openFT removes superfluous blanks.

For the FILX, ACCX and PGRX parameters, in which quotes are automatically doubled during the replacement, the number of trailing characters must be increased by the number of quotes possibly contained in the current value (i.e. 2 additional trailing characters for FILX and ACCX, whose current value can be enclosed in quotes, but which cannot themselves contain quotes, and at least 3 additional trailing characters for PGRX for the external quotes and at least one quote contained in the name).

3.9.3 Specifying a programmer's name

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

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

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

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

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

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

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

Examples

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

```
SUCC='...; //* PGRN=TEST-1'                             (1)

SUCC='...; //* PGRN=.TEST'                              (1)

SUCC='...; //* PGRN=TEST.1'                             (1)

FAIL='...; //* PGRN=''TEST 1'''                         (2)

FAIL='...; //* PGRN=''TEST/1'''                         (2)

FAIL='...; //* PGRN=''TEST1.'''                         (2)

FAIL='...; //* PGRN=''O''''REILLY'''                    (3)
```

Explanations:

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

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

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

3.10 Instance identifications

An instance ID may have a maximum length of 64 characters and may be comprised of alphanumeric characters and the special characters. You are advised only to use the special characters “.”, “-”, “:” or “%”. The first character must be alphanumeric or the special character “%”. The “%” character may only be used as a first character. An alphanumeric character must follow a “.” character.

In order to ensure the network-wide uniqueness of instance IDs, you should proceed as follows when assigning them:

– If the openFT instance has a network address with a **DNS name**, you should use this as the ID. You can create an “artificial” DNS name for an openFT instance, by placing part of a name, separated by a period, in front of an existing “neighboring” DNS name.

– If the openFT instance does not have a DNS name, but is connected to a TCP/IP network, you should use the ID **%ipn.n.n.n** (n.n.n.n is the IP address of the local openFT instance, minus the leading zeros in the address components).

The form of instance ID used internally by openFT for partners using a version earlier than V8.1, (i.e. **%.<prozessor>.<entity>)**, should not be used for your own openFT instance.

3.11 CSV output

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:

- NSTATUS
- FTSHWENV
- FTSHW
- FTSHWADS
- FTSHWKEY
- FTSHWLOG
- FTSHWMON
- FTSHWOPT
- FTSHWPTN
- FTSHWPRF
- FTSHWRGE

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.

3.12 Accounting of file transfer requests

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

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

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

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

Examples

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

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

Accounting information consists of an account number (string before the first comma), which must comply with the syntax specified above, and additional specifications with any number of characters.

openFT checks only the string before the first comma for validity with the SYS1.UADS or with RACF.

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

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

Example

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

– At the menu interface of openFT (z/OS) and the graphical user interface of openFT (BS2000), you enter this accounting information as '123$#@,ABC,12/90'.

– In an FT request issued in z/OS or BS2000, the specification is c'''123$#@,ABC,12/90'''

– At the graphical user interface of openFT for Windows or for Unix systems: '123$#@\,ABC\,12/90'

– In a command issued under Windows, enter:

- '123$#@\,ABC\,12/90'
- In a Unix system, the following must be specified in the command:
  \"123$#@\,ABC\,12/90'\'

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

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

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

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

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

Default account number

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

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

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

3.13 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.

The following ways of notifying results are available on z/OS:

- Messages and return codes automatically issued by openFT (z/OS)
- Result lists generated by openFT (z/OS)
- User-generated result information

3.13.1 Messages and return codes automatically issued by openFT (z/OS)

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

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

At the same time, a return code is entered in system variable &LASTCC ("control variable") indicating whether the NCOPY command (**not** the file transfer request) has been successful. The return codes are described in the section “Introduction to the NCOPY command” on page 350.

If a result list is requested in the NCOPY command and this result list is automatically printed out by openFT (LISTING=*SYSLST, see the operand description for LISTING in the section “Full form of the NCOPY command” on page 353), an asynchronous message indicating termination of the print job is additionally issued to the TSO user whose user ID was specified in the TRANSFER-ADMISSION for the local system. This message is generated by means of the job parameter NOTIFY in the appropriate format.

Example of a NOTIFY message:

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

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

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

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

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

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

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

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

– following successful file transfer:

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

– following unsuccessful file transfer:

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

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

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

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

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

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

3.13.2 Result lists generated by openFT (z/OS)

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

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

- Volume: the same volume as for non-existent receive files. The FT administrator can specify the volume for these (the same for all transfer requests). If the FT administrator has not specified a volume, the system defaults for newly created files apply.
- Name: <userid>.<inst>.T<transfer-id>.LST
  If the "transfer-id." consists of more than 7 characters, it is split into two parts in this name as follows:
  <userid>.<inst>.T<part1-id>.T<part2-id>.LST,
  where "part1-id" always has a length of 7 characters and "part2-id" a length of 1 to 3 characters.
  "user-id" is the user ID which was specified in the local TRANSFER-ADMISSION.

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

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

The FT administrator can predefine the basic structure of the job generated by openFT to print the result list. Further information is given in the manual "openFT (z/OS) - Installation and Operation".

3.13.3 User-generated result information

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

3.14 Notes on FTP partners

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-CR delivery unit to be installed.
- If encryption of the user data is required and the 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 encrypted if the 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.
- 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.
- 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 or z/OS. In this case, the default maximum record length applies in the receiving system.
- 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 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 openFT manual "Concepts and Functions". If a file exceeds this size, the request is cancelled with the message: “File gets no more space”.
- The 'do not overwrite' option 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 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 admission profile 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' 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.

4 openFT commands

This chapter provides a detailed description of all openFT commands in alphabetic order.

4.1 FTADDPTN
Add remote system to the partner list

Note on usage

User group: FT administrator

You can issue the FTADDPTN command under TSO.

Functional description

The FTADDPTN command is used to enter a remote system in the partner list of the local openFT instance. The network or transport system must be generated beforehand.
For details concerning the generation process, please refer to the manual "openFT (z/OS) - Installation and Operation" or another relevant manual.

The specifications which you need to enter for each partner system depend on the type of partner system (openFT (z/OS), openFT (BS2000), openFT (Unisx systems), openFT (Windows) etc.), and the method of connection to the remote system (SNA, TCP/IP directly). For details on specifying partner addresses, refer to section “Specifying partner addresses” on page 76.

If dynamic partners are permitted then inbound and outbound requests can be processed with partners which are accessed via their addresses and are not defined in the partner list.

You can issue the FTADDPTN command for all partner types while the FT system is running (openFT partner, ftp partner and ADM partner).

You can modify the partner system entry with FTMODPTN (page 243) and delete it with FTREMPTN (page 254).

Format

```
FTADDPTN

 PARTNER-NAME = <name 1..8> / *NONE
,PARTNER-ADDRESS = <text 1..200 with-low>
,SECURITY-LEVEL = *STD / *BY-PARTNER-ATTRIBUTES / <integer 1..100>
,STATE = *PARAMETERS(...)
    *PARAMETERS(...)
     |   OUTBOUND = *ACTIVE(...) / *DEACT
     |            |  *ACTIVE(...)
     |            |     AUTOMATIC-DEACT = *NO / *YES
     |  ,INBOUND = *ACTIVE / *DEACT
,IDENTIFICATION = *STD / <composed-name 1..64> / <c-string 1..64 with-low>
,SESSION-ROUTING-INFO = *NONE / *IDENTIFICATION / <alphanum-name 1..8>
,PARTNER-CHECK = *BY-FT-OPTIONS / *STD / *TRANSPORT-ADDRESS
,TRACE = *BY-FT-OPTIONS / *ON / *OFF
,AUTH-MANDATORY= *NO / *YES
,PRIORITY= *NORMAL / *LOW / *HIGH
,REQUEST-PROCESSING = *STD / *SERIAL
,RECOVERY-OUTBOUND = *BY-FT-OPTIONS / *ON / *OFF
```

Operands

PARTNER-NAME =

Symbolic name of the partner system. It can be freely selected and need only be unique within openFT.

PARTNER-NAME = <name 1..8>

The operand value “name” consists of a maximum of 8 alphanumeric characters and must be unique in the local system. The FT administrator defines this name. This name can be used in the PARTNER parameter in all FT commands in order to address the partner system.

PARTNER-NAME = *NONE

Specifies that the partner is a dynamic partner.

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

Address of the partner system. This specifies whether the partner is an openFT or FTP or ADM partner. For more information on address specifications see section “Specifying partner addresses” on page 76.

SECURITY-LEVEL =
Assigns a security level to a remote system.

SECURITY-LEVEL = *STD
If you set this operand to *STD or if you do not enter a value here, a standard security level is assigned to the remote system. This standard security level is defined using the command MODIFY-FT-OPTIONS. You can define a fixed value or specify that the value should be attribute-dependent.

SECURITY-LEVEL = *BY-PARTNER-ATTRIBUTES
If you set this operand to *STD or if you do not enter a value here, a standard security level is assigned to the remote system:

- This setting assigns partners that are authenticated by openFT the security level 10.
- Partners known to the transport system (e.g. VTAM or DNS) are assigned security level 90.
- All other partners are assigned security level 100.

SECURITY-LEVEL = <integer 1..100>
Must be specified if you wish to assign an individual security level to a specific remote system.

STATE = *PARAMETERS(...)
Controls the status of the partner system, i.e. the settings for file transfer requests issued locally (outbound) and file transfer requests issued remotely (inbound).

OUTBOUND =
Specifies the settings for file transfer requests issued locally to this partner system.

OUTBOUND = *ACTIVE(...)
File transfer requests issued locally to this partner system are processed.

AUTOMATIC-DEACT =
Defines whether cyclical attempts to establish a connection to this partner system are prohibited after a number of attempts by deactivating the partner system.

AUTOMATIC-DEACT = *NO
Failed attempts to establish a connection of this partner system do not result in its deactivation.

AUTOMATIC-DEACT = *YES
Failed attempts to establish a connection of this partner system result in its deactivation. In order to enable file transfer requests issued locally to this partner system to be executed again subsequently, it must be explicitly activated (with OUTBOUND=*ACTIVE).

OUTBOUND = *DEACT

File transfer requests issued locally to this partner system are initially not processed (not started), but are only placed in the request queue. They are executed only after the partner system has been activated with
FTMODPTN ... , STATE=(OUTBOUND=*ACTIVE).

INBOUND =

Specifies the settings for file transfer requests issued remotely, i.e. requests which are issued by this partner system.

INBOUND = *ACTIVE

File transfer requests issued remotely by this partner system are processed.

INBOUND = *DEACT

Synchronous file transfer requests issued remotely by this partner system are rejected. Asynchronous file transfer requests issued remotely by this partner system are stored there and cannot be processed until this partner system is activated with INBOUND=*ACTIVE.

IDENTIFICATION =

Network-wide, unique identification of the openFT instance in the partner system.

IDENTIFICATION = <u>*STD</u>

For openFT and FTADM partners, the partner address or the hostname from the partner address is used as the identification. For FTP partners, no identification is set.

IDENTIFICATION = <composed-name 1..64> / <c-string 1..64 with-low>

The network-wide, unique instance ID of the openFT instance in the partner system. This ID is used for authentication of partner systems as of openFT V8.1. It is set by the FT administrator of the partner system (in BS2000 by using MODIFY-FT-OPTIONS IDENTIFICATION= and in Unix systems or Windows systems by using ftmodo -id).
For more details on allocating instance IDs, please refer to section “Instance identifications” on page 88.

No instance identification may be specified for FTP partners.

SESSION-ROUTING-INFO =

If the partner system is only accessible by a go-between instance, specify the address information that the gateway instance uses for re-routing here.

SESSION-ROUTING-INFO = <u>*NONE</u>

By default, no specification is required.
The session selector can be specified as a part of the partner address.

SESSION-ROUTING-INFO = *IDENTIFICATION

Connections to the partner are re-routed via a gateway that supports the instance ID as address information.

SESSION-ROUTING-INFO = <alphanum-name 1..8>
Connections to the partner are re-routed via a gateway that supports the specified character string as address information.

PARTNER-CHECK =
Modifies the global settings for the sender check in a partner-specific way. These settings are only valid for named openFT partners that do not work with authentication.
This setting has no meaning for FTP partners and dynamic partner entries.

PARTNER-CHECK = *BY-FT-OPTIONS
The global settings are valid for the partners.

PARTNER-CHECK = *STD
Disables the expanded sender checking. The transport address of the partner is not checked, even if the expanded sender checking is globally enabled (see the FTMODOPT command).

PARTNER-CHECK = *TRANSPORT-ADDRESS
Enables the expanded sender checking. The transport address is checked, even if the expanded sender checking is globally disabled (see the FTMODOPT command).
If the transport address under which the partner is reporting does not correspond to the entry in the partner list, the request is rejected.

TRACE =
Trace setting for openFT partner systems. Trace entries are generated only when the FT trace function is activated by an operating parameter (FTMODOPT TRACE=*ON).

TRACE = *BY-FT-OPTIONS
The global settings apply for the partner.

TRACE = *ON
The trace function is activated for this partner. However, the trace is only written if the global openFT trace function is also activated (see also the FTMODOPT command, TRACE option, SWITCH=*ON). The setting made here takes priority over the setting in the operating parameters for selecting partners for the monitoring function, see the option TRACE=(...,PARTNER-SELECTION=).

i A detailed description of the trace function is provided in the manual "openFT (z/OS) - Installation and Operation".

TRACE = *OFF
The trace function is deactivated for this partner.

AUTH-MANDATORY =
Allows you to force the authentication of a named partner.

AUTH-MANDATORY = *NO
Authentication is not forced, i.e. this partner is not restricted with regard to authentication.

AUTH-MANDATORY = *YES
Authentication is forced, i.e. connections to and from this partner are only permitted with authentication.

PRIORITY=
This operand allows you to specify the priority of a partner in respect of processing requests that have the same request priority. This means that the partner priority only applies in the case of requests that have the same request priority, but that are issued to partners with a different partner priority.

PRIORITY = *NORMAL
The partner has normal priority.

PRIORITY = *LOW
The partner has low priority.

PRIORITY = *HIGH
The partner has high priority.

REQUEST-PROCESSING =
You use this option to control whether asynchronous outbound requests to this partner are always run serially or whether parallel connections are permitted.

REQUEST-PROCESSING = *STD
Parallel connections to this partner are permitted.

REQUEST-PROCESSING = *SERIAL
Parallel connections to this partner are not permitted. If multiple file transfer requests to this partner are pending, then they are processed serially. A follow-up request is consequently not started until the preceding request has terminated.

RECOVERY-OUTBOUND=
This operand controls the restart function for asynchronous outbound requests for this partner system.

RECOVERY-OUTBOUND= *BY-FT-OPTIONS
The global setting for the restart function for outbound requests is valid.

RECOVERY-OUTBOUND= *ON
The restart function for outbound requests to this partner system is activated. This is only valid if the global setting for the restart function is activated, too.

RECOVERY-OUTBOUND= *OFF
The restart function for outbound requests to this partner system is deactivated.

If the FTADDPTN command is executed correctly then no message is output.

4.1.1 Notes on entering partner systems

- You can enter the local system as a "remote" system in your own partner list. However, when performing file transfers with this system, you should note that files can be destroyed by being copied to themselves.
- It is advisable to store the FTADDPTN commands required for the entries in the partner list in a PS data set or in a PO/PDSE data set member. This facilitates the transition to a new partner list. You can generate such a file for an existing network description file using the parameter LAYOUT=*ZOS-PROC in the FTSHWPTN command.
- In large networks, especially in client-server configurations, it is a considerable effort to enter individually in the partner list all the partner systems which are to communicate with the local system. In order to reduce this effort, openFT provides with the dynamic partners option for handling file transfer and file management jobs initiated in partner systems, but which have no separate entry in the partner list (see openFT manual "Concepts and Functions").

4.1.2 Sample partner system entries

As of openFT V10 for z/OS, the file for the partner list is created when openFT is started and does not have to be created by issuing a command any longer.

The following examples demonstrate how various partner systems, accessed using a variety of transport systems, are successively entered into the partner list.

In these examples it is assumed that the local system possesses the FT identifier *ZOS1*. All partner systems that use Network Description Files or partner lists containing symbolic names for partner systems (i.e. remote openFT (z/OS) systems and openFT (BS2000) systems) address the local openFT system under the symbolic name *FTZOS1*. Although this is not really necessary from a technical point of view (symbolic names do not need to be consistent throughout the network; they simply have to be unique within the partner list on each individual openFT instance), it helps to make the examples easier to follow.

The examples below have been harmonized with other examples presented in this manual, i.e.:

- The examples relating to SNA interconnection have been harmonized with the examples relating to the generation of the transport system in the manual "openFT (z/OS) - Installation and Operation".
- The examples relating to TCP/IP interconnection have been harmonized with the example for the TNSTCPIP member of the FT parameter library, see in the manual "openFT (z/OS) - Installation and Operation".

1. A partner system with openFT as of V8.1 and the symbolic name *XAS1* is to be directly connected to the local system via TCP/IP. The instance identifier is *VAR2.MOULINET.FR*.

   If the partner's Internet address has been assigned to a host name (in the example: *XAS123*) in the z/OS name services, the remote openFT system can be entered in the local system's partner list using the following command:

   ```
   FTADDPTN PARTNER-NAME=XAS1,PARTNER-ADDRESS=XAS123
   ,IDENTIFICATION='VAR2.MOULINET.FR'
   ```

   This example functions for z/OS, BS2000, Unix and Windows partner systems, if the main station of the remote openFT system has been assigned the transport selector *$FJAM* and the port number *1100* there. These are the recommended default values in all openFT systems. Divergent values can be specified for the transport selector and the port number using the parameter PARTNER-ADDRESS in the FTADDPTN command, for example for port number 1111 and T selector TSELOPFT:

   ```
   PARTNER-ADDRESS=XAS123:1111.TSELOPFT
   ```

2. The partner system with the symbolic name *FTZOS1*, a partner with openFT V9.0 for z/OS, is to be entered in the partner list. The connection to the partner is established via SNA. It has the ftid *ZOS1* and the instance identifier *VAR1.FUSINET.AT*. The corresponding command is:

   ```
   FTADDPTN PARTNER-NAME=FTZOS1,PARTNER-ADDRESS=FJMZOS1:SNA
   ,IDENTIFICATION='VAR1.FUSINET.AT')
   ```

   In the case of partner systems with openFT V8.0 (or earlier), the instance identifier is derived from the processor name specifications and the partner system's openFT main station (usually $FJAM).

3. A partner system with openFT as of V8.1 for Unix systems and the symbolic name *FTUNIX1* is to be connected via SNA and TRANSIT-SERVER / TRANSIT-CLIENT. The LU name of the Unix partner system is *FJML0717* and its instance identifier is *UX.FUSINET.AT*.

   ```
   FTADDPTN PARTNER-NAME=FTUNIX1,PARTNER-ADDRESS=FJML0717:SNA
   ,IDENTIFICATION='UX.FUSINET.AT'
   ```

4. An FTP partner system using openFT (Unix systems) as of V10 and the symbolic name *FTPX* is to be connected over TCP/IP. The host name of the partner system is *FTPHOST1* and the default port number 21 is to be used.

   ```
   FTADDPTN PARTNER-NAME=FTPX,PARTNER-ADDRESS=FTP://FTPHOST1
   ```

5. The partner system SERVER11 with openFT V11 for Unix systems is a remote administration server. The default port number (11000) is to be used for remote administration. The partner address is to be used for identification:

   ```
   FTADDPTN PARTNER-NAME=ADMINSRV,PARTNER-ADDRESS=FTADM://SERVER11
   ```

4.2 FTADM
Execute remote administration command

Note on usage

User group: Users configured as remote administrators on the remote administration server.

A remote administration server must be deployed in order to use this command.

The command can be specified under TSO.

Description of the function

The FTADM command allows you to act as a remote administrator and administer an openFT instance via a remote administration server. The remote administration server accepts the administration request, checks the authorization and forwards the request to the openFT instance that is to be administered.

In addition, as remote administrator, you can use FTADM command to query the following information from the remote administration server (see page 113):

- You can determine what openFT instances you are authorized to administer and what remote administration permissions you have for these instances.
- You can read the ADM traps that the openFT instances you are administering have sent to the remote administration server. For this to be possible, the remote administration server must also be configured as an ADM trap server for the administered openFT instances.

Format

```
FTADM

 PARTNER-SERVER = <text 1..200 with-low>
,TRANSFER-ADMISSION = <alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...)
,ROUTING-INFO = <text 1..200 with-low> / <c-string 1..200 with-low> / *NONE
,CMD = <c-string 1..1800 with-low>
,OUTPUT =*STDERR / *STDOUT / *FILE(...)
    *FILE(...)
      |     FILE-NAME = <filename 1..59>
,DATA-ENCRYPTION = *NO / *YES
```

Operands

PARTNER-SERVER= <text 1..200 with-low>
Specifies the partner name in the partner list or the address of the remote administration server. The remote administration server must be addressed as an ADM partner. For details, see the section “Specifying partner addresses” on page 76.

TRANSFER-ADMISSION =
Specifies the FTAC transfer admission for accessing the remote administration server.

ROUTING-INFO =
Contains the routing information required to forward the remote administration command from the remote administration server to the required openFT instance.

ROUTING-INFO = <text 1..200 with-low> / <c-string 1..200 with-low>
Specifies the pathname of the openFT instance that you want to administer. The pathname is configured on the remote administration server by the ADM administrator. You can get the pathname by running the command ftshwc on the remote administration server, see the manual "openFT (z/OS) - Installation and Operation".

ROUTING-INFO = *NONE
No routing information is required, i.e. the command is executed directly on the remote administration server. Only specific commands, however, (ftshwc and ftshwatp) can be executed directly on the remote administration server. You will find a brief description of these commands on page 113.

CMD =
Remote administration server command in the syntax of the openFT instance to be administered. A remote administration command can only be processed if the remote system is using an FT product that supports this function (see the section "Remote administration commands" on page 109).

CMD = <c-string 1..1800 with-low>
The remote administration command to be executed.

OUTPUT =
Specifies where the data generated by the command should be output following transfer in the local system.

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

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

OUTPUT = *FILE(...)
The data is written to a file. Please note that only the data which the command specified with CMD outputs to *SYSLST (BS2000) or *STDOUT (on z/OS) or stdout (on a Unix/Windows system) is written to file.

FILE**-NAME = <filename 1..59>**
Name of the output file.

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 parameter.

DATA-ENCRYPTION = *NO
The data is transferred unencrypted.

DATA-ENCRYPTION = *YES
The data is transferred encrypted.

4.2.1 Remote administration commands

The following tables list the possible remote administration commands on the individual openFT platforms and on the remote administration server. The Permission column shows the permission required to execute the command as a remote administration command. The following permissions are possible:

FTOP	Read FT access (FT operator)
FT	Read and modify FT access (FT administrator)
FTAC	Read and modify FTAC access (FTAC administrator)

If a number of permissions are specified, e.g. FT **|** FTAC, it is sufficient if one of these permissions applies, i.e. FT or FTAC.

In the case of a remote administration request, these permissions are compared with the permissions you have on the relevant instance as a remote administrator. The ADM administrator defines the permissions in the configuration data of the remote administration server. If your permissions are not sufficient, the request is rejected and an appropriate message is issued.

Commands for openFT partners in BS2000

The commands have to be prefixed with "/" (slash) before the command name.

BS2000 command	**Short forms and aliases**	**Permission**
ADD-FT-PARTNER	ADD-FT-PART FTADDPTN	FT
CANCEL-FILE-TRANSFER	CAN-FILE-T, CNFT NCANCEL, NCAN FTCANREQ	FT
CREATE-FT-KEY-SET	CRE-FT-KEY FTCREKEY	FT
CREATE-FT-PROFILE	CRE-FT-PROF	FTAC
DELETE-FT-KEY-SET	DEL-FT-KEY FTDELKEY	FT
DELETE-FT-LOGGING-RECORDS	DEL-FT-LOG-REC FTDELLOG	FT | FTAC
DELETE-FT-PROFILE	DEL-FT-PROF	FTAC
IMPORT-FT-KEY [1]	IMP-FT-KEY FTIMPKEY	FT
MODIFY-FILE-TRANSFER	MOD-FILE-T FTMODREQ	FT

BS2000 command	Short forms and aliases	Permission
MODIFY-FT-ADMISSION-SET	MOD-FT-ADM	FTAC
MODIFY-FT-KEY [1]	MOD-FT-KEY FTMODKEY	FT
MODIFY-FT-OPTIONS	MOD-FT-OPT FTMODOPT	FT
MODIFY-FT-PARTNER	MOD-FT-PART FTMODPTN	FT
MODIFY-FT-PROFILE	MOD-FT-PROF	FTAC
REMOVE-FT-PARTNER	REM-FT-PART FTREMPTN	FT
SHOW-FILE-TRANSFER	SHOW-FILE-T, SHFT NSTATUS, NSTAT FTSHWREQ	FT | FTOP
SHOW-FT-ADMISSION-SET	SHOW-FT-ADM-S	FTAC
SHOW-FT-DIAGNOSTIC	SHOW-FT-DIAG FTSHWD	FT | FTOP | FTAC
SHOW-FT-INSTANCE	SHOW-FT-INST	FT | FTOP
SHOW-FT-KEY [1]	FTSHWKEY	FT | FTOP
SHOW-FT-LOGGING-RECORDS	SHOW-FT-LOG-REC FTSHWLOG	FT | FTOP | FTAC
SHOW-FT-MONITOR-VALUES [2]	SHOW-FT-MON-VAL FTSHWMON	FT | FTOP
SHOW-FT-OPTIONS	SHOW-FT-OPT FTSHWOPT	FT | FTOP
SHOW-FT-PARTNERS	SHOW-FT-PART FTSHWPTN	FT | FTOP
SHOW-FT-PROFILE	SHOW-FT-PROF	FTAC
START-FTTRACE	FTTRACE	FT | FTOP
STOP-FT	FTSTOP	FT
UPDATE-FT-PUBLIC-KEYS	UPD-FT-PUB-KEY FTUPDKEY	FT

[1] As of V12.0

[2] As of V11.0

Commands for openFT partners in z/OS

z/OS command	Alias	Permission
FTADDPTN		FT
FTCANREQ	NCANCEL, NCAN	FT
FTCREKEY		FT
FTCREPRF		FTAC
FTDELKEY		FT
FTDELLOG		FT | FTAC
FTDELPRF		FTAC
FTHELP		FT | FTOP | FTAC
FTIMPKEY [1]		FT
FTINFO		FT | FTOP | FTAC
FTMODADS		FTAC
FTMODKEY [1]		FT
FTMODOPT		FT
FTMODPRF		FTAC
FTMODPTN		FT
FTMODREQ		FT
FTREMPTN		FT
FTSHWADS		FTAC
FTSHWD		FT | FTOP | FTAC
FTSHWINS		FT | FTOP
FTSHWKEY [1]		FT | FTOP
FTSHWLOG		FT | FTOP | FTAC
FTSHWMON [2]		FT | FTOP
FTSHWNET		FT | FTOP
FTSHWOPT		FT | FTOP
FTSHWPRF		FTAC
FTSHWPTN		FT | FTOP
FTSHWREQ	NSTATUS, NSTAT	FT | FTOP
FTSTOP		FT
FTTRACE		FT | FTOP
FTUPDKEY		FT

[1] As of V12.0

[2] As of V11.0

Commands for openFT partners in Unix and Windows systems

Command	**Comment**	**Permission**
fta	up to V10.0	FT
ftaddlic	Windows systems as of V12.0 only	FT
ftaddptn		FT
ftc	up to V10.0	FT
ftcanr		FT
ftcans	openFT-Script command	FT
ftcrek		FT
ftcrep		FTAC
ftdelk		FT
ftdell		FT | FTAC
ftdelp		FTAC
ftdels	openFT-Script command	FT
fthelp		FT | FTOP | FTAC
fti	up to V10.0	FT | FTOP
ftimpk	as of V12.0	FT
ftinfo		FT | FTOP | FTAC
ftmoda		FTAC
ftmodk	as of V12.0	FT
ftmodo		FT
ftmodp		FTAC
ftmodptn		FT
ftmodr		FT
ftremlic	Windows systems as of V12.0 only	FT
ftping		FT | FTOP
ftremptn		FT
ftrs	up to V10.0	FT
ftsetpwd	Windows systems only	FT | FTOP
ftshwa		FTAC
ftshwact	openFT-Script command	FT | FTOP
ftshwd		FT | FTOP | FTAC
ftshwi		FT | FTOP
ftshwk	as of V12.0	FT | FTOP

Command	**Comment**	**Permission**
ftshwl		FT | FTOP | FTAC
ftshwlic	Windows systems as of V12.0 only	FT
ftshwm	as of V11.0	FT | FTOP
ftshwo		FT | FTOP
ftshwp		FTAC
ftshwptn		FT | FTOP
ftshwr		FT | FTOP
ftshws	openFT-Script command	FT | FTOP
ftstop		FT
fttrace		FT | FTOP
ftupdk		FT

Commands on the remote administration server

FTADM allows you to execute the commands *ftshwc* and *ftshwatp* on the remote administration server. To do this, you must specify ROUTING-INFO=*NONE:

Command	**Comment**	**Permission**
ftshwc	Gets the instances that the remote administrator is permitted to administer.	FT | FTOP | FTAC (I.e. all instances are displayed for which the remote administrator has this permission.)
ftshwatp	Outputs the ADM traps of the openFT instances that can be administered.	FT | FTOP (I.e. ADM traps of all instances are displayed for which the remote administrator has this permission.)

These commands also provide further options. For details, see, for instance, the manual "openFT (Unix and Windows systems) - Installation and Operation"".

4.3 FTCREDIR Create remote directory

Note on usage

User group: FT user

Functional description

With the FTCREDIR command, you can create a directory in an FT partner system. In remote z/OS systems you cannot create PO or PDSE dataset with the command FTCREDIR.

Format

```
FTCREDIR

 PARTNER = <text 1..200 with-low>
,DIRECTORY-NAME = *NOT-SPECIFIED / <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>
,PASSWORD = *NONE / <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>
,TRANSFER-ADMISSION = *NONE / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> /
                              *PARAMETERS(...)
   *PARAMETERS(...)
   |     USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
   |    ,ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
   |    ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
```

Operands

PARTNER = <text 1..200 with-low>
Name of the partner system as defined in the partner list by the FT administrator or the partner system address. For more information on address specifications, see section "Specifying partner addresses" on page 76.

DIRECTORY-NAME =
Name of the directory in the remote FT partner system.

DIRECTORY-NAME = *NOT-SPECIFIED
The name of the directory is known to the remote system because it has already been completely defined in the addressed FTAC admission profile, for instance.

DIRECTORY-NAME = <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>
Name of the directory in the remote system. This must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.
If the directory name is specified with an unattached Public Volume Set (BS2000 systems) then the request is rejected with error message FTR2202.

PASSWORD =
If the file system or the parent directory only permits the directory to be created with a password, you can specify this here.
This is only possible in the case of partner systems which support this type of password.

PASSWORD = *NONE
No password is required to create the directory.

PASSWORD =
<integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>
Password giving permission to create the directory in the remote system. The password must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

TRANSFER-ADMISSION =
Contains specifications concerning the transfer admission in the remote system for the file management request.

TRANSFER-ADMISSION = *NONE
The remote system does not require or does not know any user admissions.

TRANSFER-ADMISSION =
<alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
If FTAC functionality is used in the remote system then the transfer admission for the remote system can be defined via an admission profile. In this case, only the TRANSFER-ADMISSION defined in the admission profile is used here. The alphanumeric input is converted to lowercase internally.

TRANSFER-ADMISSION = *PARAMETERS(...)
Specifies the user's identification, account number and password in the remote system. The operands in the brackets can also be used as positional operands without the associated keywords.

USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
Identification of the user in the remote system. The identification must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
Account number of the user in the remote system. The account number must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

PASSWORD =
Password allowing the user to access the remote system.

PASSWORD = <u>*NONE</u>
Access is possible without a password.

PASSWORD =
<c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
Password allowing the user to access the remote system. The password must be specified in the syntax of the remote system, must adhere to the conventions used in the remote system and must be known there.

Example:

The directory `Dir1` is to be created in the remote Unix system under the user ID with the transfer admission `transadm`:

```
ftcredir partux,c'Dir1',,transadm
```

4.4 FTCREKEY Create a key pair set

Note on usage

User group: FT administrator

You can issue the FTCREKEY command under TSO with the FT system running.

Functional description

Using this FTCREKEY command, you create a key pair for authenticating your openFT instance in partner systems (RSA procedures). The key pair consists of a private key, administered internally by openFT, and a public key.

Public keys are stored under the name:

<openft qualifier>.<inst>.SYSPKF.R<key reference>.L<key length>

Here, the first two name parts are replaced by OPENFT QUALIFIER and the instance name.

The key reference is a numerical designator for the version of the key pair. The key length is 768 or 1024 or 2048. The three key lengths are always generated. The public key files are text files which are created in the character code of the respective operating system, i.e. EBCDIC.DF04-1 for BS2000, IBM1047 for z/OS, ISO8859-1 for Unix systems and CP1252 for Windows systems.

In a file <openft qualifier>.<inst>.SYSPKF.COMMENT you can store comments, which are written in the first lines of the public key files when a key pair set is created. Such comments could be, for example, the communications partner and the telephone number of the FT administrator on duty. The lines in the SYSPKF.COMMENT file may be a maximum of 78 characters long.

So that your openFT instance can be authenticated by partner systems (using openFT as of version 8.1), the public key file must be transported to the partners via a reliable path and re-coded if necessary.

In order to make an authorized update of the key pair sets, openFT supports up to three key pair sets at a time.

The most current key pair is used for delivering the session key for encrypting user data and request description data. If there is no key pair set, work proceeds without encryption.

Format

FTCREKEY

Without operands

In the event of an error (three key pair sets already exist), the following message is output:

```
FTR1029 OPENFT: Maximum number of key pairs exceeded
```

4.5 FTCREPRF Create admission profile

Note on usage

User group: FTAC user and FTAC administrator

A prerequisite for using this command is the use of openFT-AC.

Functional description

All FTAC users can use FTCREPRF to set up their own admission profiles under their user IDs. Users must activate admission profiles predefined by the FTAC administrator with FTMODPRF (see page 220) before they can be used. Profiles predefined by the FTAC administrator may be used immediately if the FTAC administrator also possesses the SU privilege.

The FTAC administrator can use FTCREPRF to create admission profiles for each user. It is necessary to distinguish between three cases:

- The FTAC administrator possesses theSU privilege. He/She can then create profiles for other user IDs without restriction which are available for immediate use if they are complete. If the FTAC administrator specifies *NOT-SPECIFIED for PASSWORD in the USER-ADMISSION operand, the profiles are not locked, but they cannot be used, either.
- If the FTAC administrator does not possess the SU privilege but specifies ACCOUNT and PASSWORD in the USER-ADMISSION parameter, then he/she may also assign a TRANSFER-ADMISSION for the profile. However, this functions only for as long as the current password for the user ID corresponds to the one defined in the profile.
- If the FTAC administrator does not possess the SU privilege and also does not specify the user's account number and password, then he/she may not define any TRANSFER-ADMISSION in the profile. In this case, the user must then assign the profile a TRANSFER-ADMISSION with the FTMODPRF command, and the specifications for the USER-ADMISSION must, if necessary, be complemented.

  Example

  The FTAC administrator creates an admission profile for user USER1. In doing so he/she specifies only the user ID for the USER-ADMISSION, but not the account number and password. In this case the FTAC administrator may also not specify a TRANSFER-ADMISSION.

  ```
  CR-FT-PROF NAME=HISPROF2,TRANS-ADM=*NOT-SPECIFIED, -
  USER-ADM=(USER1,*NOT-SPECIFIED,*NOT-SPECIFIED)
  ```

– It is possible to create an admission profile for "preprocessing" or "postprocessing". To do this, the FILE-NAME operand must start with the pipe symbol '|'. After this has been done, one or more TSO commands can be specified. For detailed information refer to the section “Preprocessing and postprocessing” on page 81.

Format

(part 1 of 2)

```
FTCREPRF

 NAME = *STD / <alphanum-name 1..8>
,PASSWORD = *NONE / <alphanum-name 1..8>
,TRANSFER-ADMISSION = *NOT-SPECIFIED / <alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) /
                      <x-string 15..64>(...)

   <alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...)
        VALID = *YES / *NO
       ,USAGE = *PRIVATE / *PUBLIC
       ,EXPIRATION-DATE = *NOT-RESTRICTED / <date 8..10>
,PRIVILEGED = *NO / *YES
,IGNORE-MAX-LEVELS = *NO / *YES / *PARAMETERS(...)

   *PARAMETERS(...)
        OUTBOUND-SEND = *NO / *YES
       ,OUTBOUND-RECEIVE = *NO / *YES
       ,INBOUND-SEND = *NO / *YES
       ,INBOUND-RECEIVE = *NO / *YES
       ,INBOUND-PROCESSING = *NO / *YES
       ,INBOUND-MANAGEMENT = *NO / *YES
,USER-ADMISSION = *OWN / *PARAMETERS(...)

   *PARAMETERS(...)
        USER-IDENTIFICATION = *OWN / <name 1..8>
       ,ACCOUNT = *OWN / *NOT-SPECIFIED / *NONE / <alphanum-name 1..40> / <c-string 1..40>
       ,PASSWORD = *OWN / *NOT-SPECIFIED / <alphanum-name 1..8> / *NONE
,INITIATOR = (*LOCAL, *REMOTE) / list-poss(2): *LOCAL / *REMOTE
,TRANSFER-DIRECTION = *NOT-RESTRICTED / *FROM-PARTNER / *TO-PARTNER
,PARTNER = *NOT-RESTRICTED / list-poss(50): <text 1..200 with-low>
,MAX-PARTNER-LEVEL = *NOT-RESTRICTED / <integer 0..100>
```

(part 2 of 2)

,**FILE-NAME** = ***NOT-REST**RICTED / <filename1..59> / <c-string 1..512 with-low> /
***EXP**ANSION(...)

,***EXP**ANSION(...)
  PREFIX = <filename 1..58> / <filename-prefix 2..50> /
  <c-string 1..511 with-low>

,**FILE-PASS**WORD = ***NOT-REST**RICTED / ***NONE** / <alphanum-name 1..8>

,**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..40> / <c-string 1..40>
  ,**PASSW**ORD = ***SAME** / ***NOT-REST**RICTED / ***NONE** / <alphanum-name 1..8>

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

***EXP**ANSION(...)
  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 / ***REM**OTE**-ADM**INISTRATION

,**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 on the user ID specified in USER-ADM. If an admission profile with this name already exists, FTAC rejects the command with the message:

```
FTC0100 COMMAND REJECTED. FT-PROFILE ALREADY EXISTS
```

The command FTSHWPRF (see page 321) can be used to view the already existing names. To obtain this information, the command FTSHWPRF can be entered without operands and a user ID must be specified.

NAME = *STD
Creates a standard admission profile for the user ID. You must specify *NOT-SPECIFIED as the transfer admission, because a standard 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 standard admission profile.

PASSWORD =
FTAC password which authorizes you to issue FTAC commands on your user ID, if such a password was defined in your admission set.

PASSWORD = *NONE
No FTAC password is required.

PASSWORD = <alphanum-name 1..8>
This FTAC password is required.

TRANSFER-ADMISSION =
With TRANSFER-ADMISSION, you define the transfer admission. If this transfer admission is entered in an FT request instead of the LOGON authorization, then the access rights are valid which are defined in this admission profile. This transfer admission must be unique in the entire openFT instance, so that there is no conflict with other transfer admissions which other FTAC users have defined for other access rights. When the transfer admission which you have selected has already been used, then FTAC rejects the command with the message:

```
FTC0101 COMMAND REJECTED. TRANSFER-ADMISSION ALREADY EXISTS
```

The FTAC administrator can also assign a transfer admission when he/she creates an admission profile for a user ID. If the FTAC administrator possesses no SU privilege, he/she must also enter the complete USER-ADMISSION for the user ID in question (USER-IDENTIFICATION, ACCOUNT and PASSWORD).

TRANSFER-ADMISSION = *NOT-SPECIFIED
This entry is used to set up a profile without transfer admission. If the profile is not a standard admission profile, it is locked until you specify a valid transfer admission or the owner specifies 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. 2017-12-31 or 17-12-31 for December 31, 2017. The use of the transfer admission is only possible until the given date.

PRIVILEGED =

The FTAC administrator can privilege the profile. FT requests which are processed with a privileged admission profile are not subject to the restrictions which are set for MAX-ADM-LEVEL (see page 268) 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.

PRIVILEGED = *YES

The admission profile is privileged.
Only the FTAC administrator can use this entry.

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 268).

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.
If, as FTAC administrator, you create the admission profile for a user, you cannot generally specify neither ACCOUNT nor PASSWORD in the USER-ADMISSION operand (since these should be known only to the user in question). These specifications must be entered by the user by means of FTMODPRF before the profile can actually be used.

Please observe the note on PASSWORD=*OWN on page 127.
As FTAC administrator you can create a profile which is available for immediate use, i.e. a profile with the TRANSFER-ADMISSION defined, only if you specify the USER-ADMISSION with ACCOUNT and PASSWORD or if you also possess the SU privilege. For ACCOUNT= you can also specify *NOT-SPECIFIED or *NONE.

USER-ADMISSION = *OWN
For USER-IDENTIFICATION and ACCOUNT, the specifications are taken from the current LOGON authorization. A possible z/OS password is only taken from your LOGON authorization when an FT request accesses the admission profile. This specification consequently generates a profile in the current user ID.

USER-ADMISSION = *PARAMETERS(...)
Specifies the individual components of the user ID.
This allows you to keep FT requests which use this admission profile under an account number other than the current one, for example. Or, a password can be set in the admission profile. FT requests which use this admission profile will then only function if the current LOGON password corresponds to the preset password.

USER-IDENTIFICATION =
User ID in z/OS.

USER-IDENTIFICATION = *OWN
The user ID is taken from the current LOGON authorization.

USER-IDENTIFICATION = <name 1..8>
User ID to which the profile should belong. As FTAC administrator you may also specify foreign user IDs. As an FTAC user you can only specify your own user ID; the specification corresponds to *OWN.

ACCOUNT =
Account number under which an FT request is to be kept when it uses this admission profile.

ACCOUNT = *OWN
The account number is taken from the current LOGON authorization.

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

ACCOUNT = *NONE
Has the same effect as ACCOUNT = *NOT-SPECIFIED.

ACCOUNT = <alphanum-name 1..8>
An FT request should be kept under the account number specified when it accesses this admission profile. You can enter any account number which belongs to the user ID specified in the USER-IDENTIFICATION.
You can also specify accounting information which contains the account number to be used.

PASSWORD =
z/OS password which an FT request should use when it works with this admission profile.

PASSWORD = *OWN
When an FT request refers to this admission profile, FTAC uses the BS2000 password valid for at that moment. This prevents you from having to modify the admission profile if the BS2000 password is changed.

Admission profiles in which PASSWORD is set to its default value via *OWN cannot be used for preprocessing, postprocessing or follow-up processing. For preprocessing and postprocessing, the password must be explicitly assigned a value in USER-ADMISSION. For follow-up processing, a specification in PROCESSING-ADMISSION is also possible.

PASSWORD = *NOT-SPECIFIED
The password will be entered by the owner of the admission profile. This function allows the FTAC administrator to create profiles for foreign user IDs whose access data he/she does not know.

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

PASSWORD = <alphanum-name 1..8>
When an FT request accesses the admission profile, the password specified is compared with the current LOGON password. If the two do not correspond, the FT request is rejected.

INITIATOR =
Determines if initiators from local and/or remote systems are permitted to use this admission profile for their FT requests.

INITIATOR = (*LOCAL,*REMOTE)
This admission profile may be used by initiators from local and remote systems.

INITIATOR = *REMOTE
This admission profile may only be used for FT requests by initiators from remote systems.

INITIATOR = *LOCAL
This admission profile may only be used for FT requests by initiators from the local system.

TRANSFER-DIRECTION =
Determines which transfer direction may be used with this admission profile. The transfer direction is always determined from the system in which the admission profile was defined.

TRANSFER-DIRECTION = *NOT-RESTRICTED
With this admission profile, files can be transferred to and from a partner system.

TRANSFER-DIRECTION = *FROM-PARTNER
With this admission profile, files can only be transferred from a partner system to your system. It is not possible to display file attributes/directories (partial components of “inbound file management”).

TRANSFER-DIRECTION = *TO-PARTNER
With this admission profile, files can only be transferred from your system to a partner system. It is not possible to modify file attributes or delete files (partial components of “inbound file management”).

PARTNER =
Specifies that this admission profile is to be used only for FT requests which are processed by a 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 “Specifying partner

addresses” on page 76. 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.

i When you set MAX-PARTNER-LEVEL=0, you prevent access to the admission profile (for the moment). No FT requests can be processed with this admission profile.

FILE-NAME =
Determines which files or library members under your user ID may be accessed by FT requests that use this admission profile.

FILE-NAME = *NOT-RESTRICTED
Permits unrestricted access to all files and library members of the user ID.

FILE-NAME = <filename 1..59> / <c-string 1..512 with-low>
Only the specified file may be accessed. However, openFT is also able to generate unique filenames automatically, thus providing an easy way of avoiding conflicts. This is done by specifying the string %UNIQUE at the end of the filename which is predefined here. When follow-up processing is specified, this file can be referenced with %FILENAME, %FILN or %FILX.
You can also directly specify file transfer with file preprocessing or postprocessing here by entering a pipe symbol ’|’ followed by TSO commands.

FILE-NAME = *EXPANSION(PREFIX = <filename 1..58> / <partial-filename 2..50> / <c-string 1..511 with-low>)
Restricts access to a number of files which all begin with the same prefix. If a *filename* is entered in an FT request which works with this admission profile, FTAC sets the *prefix* defined with EXPANSION in front of this filename. The FT request is then permitted to access the file *PrefixFilename*.

Example

- PREFIX=JACK.; an FT request in which FILE-NAME=BOERSE is specified, then accesses the file JACK.BOERSE.
- PREFIX=TOOLS.CLIST/; an FT request in which FILE-NAME=MEMBER01 is specified, then accesses the file TOOLS.CLIST(MEMBER01).

Please note that the part of a filename which is specified in the file transfer command still has to be of the type <filename>.

If you want to perform file transfer with pre- or postprocessing, 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 postprocessing 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 the section section “Preprocessing and postprocessing” on page 81.

The maximum length of the entire pre- or postprocessing command is limited to the maximum length of the file name. If several commands are specified, then they must be separated by a semicolon (‘;’).

Example

```
FILE-NAME = *EXP(C‘|Command1;Command2;Command3; ...‘)
```

If you specify a name prefix that starts with a pipe character with *EXP(PREFIX=...), the preprocessing or postprocessing command of the FT request must not contain any semicolons. If the preprocessing or postprocessing command nevertheless contains semicolons, it must be enclosed in '...' (single quotes) .

Special cases

- A file name or file name prefix that begins with the string '|ftexecsv' must be specified for admission profiles that are to be exclusively used for the ftexec command (see page 138).
- 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 page 137).

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 = <u>*NOT-RESTRICTED</u>
Permits access to all files. If a password is set for a file, then it must be specified in the transfer request.

FILE-PASSWORD = *NONE
Only permits access to files without file passwords.

FILE-PASSWORD = <alphanum-name 1..8>
Only permits access to files which are protected with the password specified and to unprotected files. The password which has already been specified in the profile may not be repeated in the transfer request. PASSWORD=*NONE would be entered in this case!

PROCESSING-ADMISSION =
You can enter a user ID in your z/OS system. Any follow-up processing of an FT request will be executed under this user ID. With PROCESSING-ADMISSION in the admission profile, you do not need to disclose your LOGON authorization to partner systems for follow-up processing.

i Admission profiles in which ACCOUNT and/or PASSWORD in USER-ADMISSION are set to their default values via *OWN cannot be used for follow-up processing. For follow-up processing, these parameters must be explicitly assigned a value either in USER-ADMISSION or in PROCESSING-ADMISSION.

PROCESSING-ADMISSION = <u>*SAME</u>
For the PROCESSING-ADMISSION, the values of the USER-ADMISSION are used. If *SAME is entered here, then any FT request which uses this profile must also contain PROCESSING-ADMISSION=*SAME or PROCESSING-ADMISSION=*NOT-SPECIFIED.

PROCESSING-ADMISSION = *NOT-RESTRICTED
FT requests which use this admission profile may contain any PROCESSING-ADMISSION.

PROCESSING-ADMISSION = *PARAMETERS(...)
You can also enter the individual components of the user ID. This allows you to keep FT requests which use this admission profile under a different account number, for example. Or, a password can be set in the admission profile. FT requests which use this admission profile will then only function if their current LOGON password corresponds to the pre-set password.

USER-IDENTIFICATION =
Identifies the user ID under which the follow-up processing is to be executed.

USER-IDENTIFICATION = <u>*SAME</u>
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 admission profile is used.

ACCOUNT = <alphanum-name 1..40> / <c-string 1..40>
Follow-up processing is to be settled under this account number.
You an also specify accounting information containing the account number to be used.

PASSWORD =
You specify, where applicable, the z/OS password for the user ID specified in the USER-IDENTIFICATION under which the follow-up processing is to be executed. Here, you can enter a PASSWORD when the user ID in question doesn't have such a password (yet).

PASSWORD = *SAME
The value *SAME is only valid if the PROCESSING-ADMISSION refers to your own user ID. If PASSWORD=*OWN is entered on USER-ADMISSION, then the password valid at the time of the request is used for the PROCESSING-ADMISSION.

PASSWORD = *NOT-RESTRICTED
Specifies the password in FT requests which work with the admission profile. The admission profile does not restrict the password with regard to follow-up processing.

PASSWORD = *NONE
FT requests which use this admission profile can only initiate follow-up processing on user IDs without a password.

PASSWORD = <alphanum-name 1..8>
FT requests which use this admission profile may only initiate follow-up processing on user IDs which are protected with this password.

SUCCESS-PROCESSING =
Restricts the follow-up processing which an FT request is permitted to initiate in your system after a successful data transfer.

SUCCESS-PROCESSING = *NOT-RESTRICTED
In FT requests which use this admission profile the operand SUCCESS-PROCESSING may be used without restriction.

SUCCESS-PROCESSING = *NONE
The admission profile does not permit follow-up processing after successful data transfer.

SUCCESS-PROCESSING = <c-string 1..1000 with-low>
Commands which are executed in the local system after successful data transfer.
The individual commands must be separated by a semicolon (;). If a character string is enclosed by single or double quotes (’ or ”) within a command sequence, openFT does not interpret any semicolons within this character string as a separator.

SUCCESS-PROCESSING = *EXPANSION(...)
If a SUCCESS-PROCESSING was specified in an FT request which uses this admission profile, FTAC adds the prefix or suffix specified here to this command. As follow-up processing, the command which has been thus expanded is then executed.

If a suffix or prefix is defined at this point, then no command sequence for the follow-up processing may be specified in FT requests which use this admission profile. This makes the setting of prefixes and suffixes mandatory.

PREFIX = *NOT-RESTRICTED
Follow-up processing is not restricted by a prefix.

PREFIX = <c-string 1..999 with-low>
The specified prefix is set in front of a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the prefix is executed as follow-up processing.

SUFFIX = *NOT-RESTRICTED
The follow-up processing is not restricted by a suffix.

SUFFIX = <c-string 1..999 with-low>
The specified suffix is added to a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the suffix is executed as follow-up processing.

Note that blanks at the end of the specification are removed in the FT request, when the follow-up command is assembled. Therefore blanks that are needed here, must be included at the beginning of the specification for SUFFIX.

Example

If PREFIX='SEND ' and SUFFIX=',USER(USER1)' is specified and SUCC="'FILE TRANSFER OK'" is defined in the FT request, FT executes the command "SEND 'FILE TRANSFER OK',USER(USER1)" for follow-up processing.

FAILURE-PROCESSING =
Restricts the follow-up processing which an FT request is permitted to initiate in your system after a failed data transfer.

FAILURE-PROCESSING = *NOT-RESTRICTED
In FT requests which use this admission profile the operand FAILURE-PROCESSING may be used without restriction.

FAILURE-PROCESSING = *NONE
The admission profile does not permit follow-up processing after failed data transfer.

FAILURE-PROCESSING = <c-string 1..1000 with-low>
Commands which are executed in the local system after failed data transfer.
The individual commands must be separated by a semicolon (;). If a character string is enclosed by single or double quotes (’ or ”) within a command sequence, openFT does not interpret any semicolons within this character string as a separator.

FAILURE-PROCESSING = *EXPANSION(...)
If a FAILURE-PROCESSING was specified in an FT request which uses this admission profile, FTAC adds the prefix or suffix specified here to this command. As follow-up processing, the command which has been thus expanded is then executed.

If a suffix or prefix is defined at this point, then no command sequence for the follow-up processing may be specified in FT requests which use this admission profile. This makes the setting of prefixes and suffixes mandatory.

PREFIX = *NOT-RESTRICTED
Follow-up processing is not restricted by a prefix.

PREFIX = <c-string 1..999 with-low>
The specified prefix is set in front of a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the prefix is executed as follow-up processing.

SUFFIX = *NOT-RESTRICTED
The follow-up processing is not restricted by a suffix.

SUFFIX = <c-string 1..999 with-low>
The specified suffix is added to a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the suffix is executed as follow-up processing.

WRITE-MODE =
Determines the WRITE-MODE specification which is valid for this FT request. WRITE-MODE is only effective if the receive file is in the same system as the admission profile definition.

WRITE-MODE = *NOT-RESTRICTED
In an FT request which accesses this admission profile, the operand WRITE-MODE may be used without restrictions.

WRITE-MODE = *NEW-FILE
In the FT request, *NEW-FILE, *REPLACE-FILE or *EXTEND-FILE may be entered for WRITE-MODE. If the receive file already exists, the transfer will be rejected.

WRITE-MODE = *REPLACE-FILE
In the FT request of openFT partners, only *REPLACE-FILE or *EXTEND-FILE may be entered for WRITE-MODE. With ftp partners, *NEW-FILE may also be entered if the file does not yet exist.

WRITE-MODE = *EXTEND-FILE
In the FT request, only *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).

FT-FUNCTION = *NOT-RESTRICTED
The full scope of FT functions is available. For reasons of compatibility, the specification NOT-RESTRICTED means that FILE-PROCESSING and REMOTE-ADMINISTRATION are not permitted! All other functions are permitted if this value is specified.

FT-FUNCTION = (*TRANSFER-FILE, *MODIFY-FILE-ATTRIBUTES, *READ-DIRECTORY,*FILE-PROCESSING, *REMOTE-ADMINISTRATION)
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 “preprocessing” and “postprocessing” 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 preprocessing and postprocessing (pipe symbol |) are to be possible with this FTAC profile.

***REMOTE-ADMINISTRATION**
The admission profile is allowed to be used for the "remote administration" function. This allows a remote administrator to administer the openFT instance using this profile. *REMOTE-ADMINISTRATION may only be specified by the FT administrator or FTAC administrator.

USER-INFORMATION =
Here, you enter a text in the admission profile. This text is displayed with the command FTSHWPRF.

USER-INFORMATION = <u>*NONE</u>
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 = <u>*NOT-RESTRICTED</u>
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.

i When using restrictions for FILE-NAME, SUCCESS-PROCESSING and FAILURE-PROCESSING, keep in mind that

- a restriction for follow-up processing must always be made for SUCCESS- and FAILURE-PROCESSING. Otherwise, it is possible that users will avoid this step.
- PREFIX of FILE-NAME, SUCCESS-PROCESSING and FAILURE-PROCESSING must correspond,
  e.g. FILE-NAME = *EXP(XYZ.),SUCC = *EXP('PR DSNAME(XYZ.',')')

Examples

1. Jack John wishes to create an admission profile for the following purpose:

Dylan Dack, employee at the Dack Goldmine, has his own z/OS computer. He has to transfer monthly reports on a regular basis to his boss Jack‘s computer, JACKJOHN, using File Transfer. The file needs to have the name MONTHLY.REPORT.GOLDMINE and is to be printed out after transfer.
The JCL statement for printing out the file MONTHLY.REPORT.GOLDMINE is completely contained in the member GOLDMOBE of the PO data set PRINT.

Since Jack’s admission set does not permit any “inbound” requests, he needs to give the profile privileged status (he/she is permitted to do this, since he is an FTAC administrator). The Goldmine computer has the security level 50. The command required to create such an admission profile is as follows:

```
 FTCREPRF NAME=GOLDMOBE,                                 -
          TRANSFER-ADMISSION=MONTHLYREPORTFORTHEBOSS,    -
          PRIVILEGED=*YES,                               -
          IGNORE-MAX-LEVELS=*YES,                        -
          USER-ADM=(STEFAN,XXXX,PASSWD),                 -
          TRANSFER-DIRECTION=*FROM-PARTNER,              -
          PARTNER=GOLDMINE,                              -
          FILE-NAME=MONTHLY.REPORT.GOLDMINE,             -
          SUCCESS-PROCESSING=                            -
          'ALLOC DSNAME(PRINT(MONTHLY.REPORT.GOLDMINE))',-
          FAILURE-PROCESSING=*NONE,                      -
          WRITE-MODE=*REPLACE-FILE
```

The short form of this command is:

```
FTCREPRF␣GOLDMOBE,TRANS-AD=MONATSBERICHTFUERDENCHEF,            -
IGN-MAX-LEV=*YES,USER-ADM=(STEFAN,XXXX,PASSWD),                 -
PRIV=*YES,IGN-MAX-LEV=*YES,USER-ADM=(STEFAN,XXXX,PASSWD),       -
TRANS-DIR=*FROM,PART=GOLDMINE,                                  -
FILE-NAME=MONATS.BERICHT.GOLDMINE,                              -
SUCC='ALLOC DSNAME(PRINT(MONATS.BERICHT.GOLDMINE))',FAIL=*NONE, -
WRITE=*REPL
```

File management can also be performed with this admission profile (see the specifications for the IGNORE-MAX-LEVELS operand).

Dylan Dack, who keeps the monthly report for the goldmine in his z/OS computer in the file NOTHING.BUT.LIES, can use the following openFT command to send it to the central computer JACKJOHN and print it out there:

```
/NCOPY␣TO,JACKJOHN,(NOTHING.BUT.LIES), -
       REM=*MSP(FILE=*NOT-SPECIFIED,TRANS-AD=MONTHLYREPORTFORTHEBOSS)
```

2. A profile is to be created that only allows monitoring.

```
FTCREPRF MONITOR,,'ONLYFTMONITOR'  -
   ,FILE-NAME=*EXP('|*FTMONITOR ') -
   ,FT-FUN=(*TRANS-F,*FILE-PROC)
```

The openFT Monitor can be started from a Unix or Windows system using this profile with the following command:

```
ftmonitor "-po=10" FTZOS ONLYFTMONITOR
```

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

```
ncopy FTZOS!"-po=10" ftzos_data ONLYFTMONITOR
```

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.

4. You want to create a profile which can be used to run precisely one file processing command. A number of logging records are output in the example below.

   ```
   FTCREPRF NUR1VORV,,'GetLoggingRecords'             -
   ,USER-ADMISSION=(STEFAN,xxxx,password)             -
   ,FILE-NAME=*EXP('|ftexecsv -p="FTSHWLOG ,"')       -
   ,FT-FUN=(*TRANS-F,*FILE-PROC)
   ```

   The following command, for example, can be used to access the profile from a remote system:

   – Unix system or Windows system:

     ```
     ftexec FTZOS 3 GetLoggingRecords
     ```

   – BS2000 system:

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

   – z/OS system:

     ```
     FTEXEC FTZOS,'3','GetLoggingRecords'
     ```

4.6 FTDEL Delete remote files

Note on usage

User group: FT user

Functional description

The FTDEL command can be used to delete a file in an FT partner system. You cannot delete directories with this command including PO and PDSE data sets.

Format

```
FTDEL
------------------------------------------------------------------------------------------------
 PARTNER = <text 1..200 with-low>
,FILE = *NOT-SPECIFIED / <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>
,PASSWORD = *NONE / <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>
,TRANSFER-ADMISSION = *NONE / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> /
                             *PARAMETERS(...)
   *PARAMETERS(...)
    |  USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
    | ,ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
    | ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
```

Operands

PARTNER = <text 1..200 with-low>
Name of the partner system as defined in the partner list by the FT administrator or the partner system address. For more information on address specifications, see section “Specifying partner addresses” on page 76.

FILE =
The name of the file in the remote FT partner system.

FILE = *NOT-SPECIFIED
The name of the file is known to the remote system because it has already been completely defined in the addressed FTAC admission profile, for instance.

FILE = <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>
The name of the file in the remote system. The file name must be specified in the syntax of the remote system and must conform to the conventions of the remote system.

PASSWORD =
The password that provides access to the file in the remote system. If the file in the remote system is password-protected, the password required for deleting files in the remote system must be specified in these operands.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>
The password that provides access to the file in the remote system. The password must be specified in the syntax of the remote system and conform to the conventions of the remote system.

TRANSFER-ADMISSION =
Contains specifications on transfer admission to the remote system for file management requests.

TRANSFER-ADMISSION = *NONE
The remote system does not require or recognize user authorization.

TRANSFER-ADMISSION =
<alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
The transfer admission for the remote system can only be defined in an admission profile if the FTAC functionality is in use in the remote system. In this case, only the TRANSFER-ADMISSION defined in the FT profile is specified here. Uppercase alphanumeric input is converted internally to lowercase.

TRANSFER-ADMISSION = *PARAMETERS(...)
Specifies the ID, the account number, and the password of the user in the remote system. The operands in brackets can also be used as positional operands without their keywords.

> **USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>**
> User ID in the remote system. The ID must be specified in the syntax of the remote system and must conform to the conventions of the remote system.
>
> **ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>**
> Account number for the user in the remote system. The account number must be specified in the syntax of the remote system and must conform to the conventions of the remote system.
>
> **PASSWORD =**
> The password that allows the user to access the remote system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
The password that allows the user to access the remote system. The password must be specified in the syntax of the remote system, must conform to the conventions of the remote system, and be recognized by the remote system.

Example

From your z/OS system, you want to delete the file FILE which is stored in the partner system HUGO. FTAC is implemented in the remote system. The transfer admission DELETEACCESS must be specified to delete the file.

```
FTDEL PARTNER=HUGO,FILE=FILE,TRANSFER-ADMISSION=DELETEACCESS
```

Short form:

```
FTDEL HUGO,FILE,,DELETEACCESS
```

4.7 FTDELDIR Delete remote directory

Note on usage

User group: FT user

Functional description

With the FTDELDIR command, you can delete an empty directory in an FT partner system. A directory means:

- a PO or PDSE dataset or an openEdition directory in a z/OS system
- a PLAM library in a BS2000 system or a directory in a POSIX file system
- any directory in an Unix system or Windows system

PO or PDSE datasets or a PLAM libraries also can be deleted, if they are not empty.

Format

```
FTDELDIR
--------------------------------------------------------------------------------------------------
 PARTNER = <text 1..200 with-low>
,DIRECTORY-NAME = *NOT-SPECIFIED / <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>
,PASSWORD = *NONE / <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>
,TRANSFER-ADMISSION = *NONE / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> /
                      *PARAMETERS(...)
  *PARAMETERS(...)
    |  USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
    | ,ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
    | ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19> /
    |
```

Operands

PARTNER = <text 1..200 with-low>
Name of the partner system as defined in the partner list by the FT administrator or the partner system address. For more information on address specifications, see section “Specifying partner addresses” on page 76.

DIRECTORY-NAME =
Name of the directory in the remote FT partner system.

DIRECTORY-NAME = *NOT-SPECIFIED
The name of the directory is known to the remote system because it has already been completely defined in the addressed FTAC admission profile, for instance.

DIRECTORY-NAME = <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>
Name of the directory in the remote system. This must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.
If the directory name is specified with an unattached Public Volume Set (BS2000 systems) then the request is rejected with error message FTR2202.

PASSWORD =
Password making it possible to access the directory in the remote system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>
Password allowing the user to delete the directory in the remote system. The password must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

TRANSFER-ADMISSION =
Contains specifications concerning the transfer admission in the remote system required to execute the file management request.

TRANSFER-ADMISSION = *NONE
The remote system does not require or does not know any user admissions.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
If FTAC functionality is used in the remote system then the transfer admission for the remote system can be defined via an admission profile. In this case, only the TRANSFER-ADMISSION defined in the admission profile is used here. In the case of alphanumeric input, uppercase is converted to lowercase internally.

TRANSFER-ADMISSION = *PARAMETERS(...)
Specifies the user's identification, account number and password in the remote system. The operands in the brackets can also be used as positional operands without the associated keywords.

USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
Identification of the user in the remote system. The identification must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
Account number of the user in the remote system. The account number must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

PASSWORD =
Password allowing the user to access the remote system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
Password allowing the user to access the remote system. The password must be specified in the syntax of the remote system, must adhere to the conventions used in the remote system and must be known there.

Example:

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

```
ftdeldir partux,c'Dir1',,transadm
```

4.8 FTDELKEY Delete a key pair set

Note on usage

User group: FT administrator

The command can only be specified under TSO.

Functional description

Using the FTDELKEY command, you are deleting the key pair set of a reference. The key pair consists of a private key, which is internally administered by openFT, and a public key.

Public keys are stored under:

<openft qualifier>.<inst>.SYSPKF.R<key reference>.L<key length>

Here, the first two name parts are replaced by OPENFT QUALIFIER and the name of the instance.

The key reference is a numeric designator for the version of the key pair. For each reference there are three keys with lengths of 768, 1024 and 2048 bits respectively.

A key pair set should only be deleted if no partner system uses the corresponding public key any longer. This means that, after creating a new key pair set using FTCREKEY, the new public key should be made available to all of the partner systems in which the local system is to be authenticated.

There should always be at least one key pair set in your openFT instance, otherwise all requests will be carried out in unencrypted form.

Format

FTDELKEY
REFERENCE = <integer 1..9999999>

Operands

REFERENCE = <integer 1..9999999>
Allows selection of the key pair set to be deleted. You will find the reference in the name of the public key file (see above).

4.9 FTDELLOG Delete log records or offline log files

Note on usage

User group: FT administrator, FTAC administrator

The command can be entered under TSO.

Functional description

With FTDELLOG you can, as FT or FTAC administrator, delete log records for all login names and all record types (FT, FTAC, ADM) from the current log file.

You can also delete offline log files which are no longer required. Offline log files can only be deleted in their entirety. It is not possible to delete individual log records from an offline log file.

In principle, openFT can write any number of logging records (until the disk is full). The FT administrator should save the existing logging records (e.g. to tape or as a file in CSV format) and at regular intervals (weekly, for example, if there is a large number of requests) and delete older logging records. This means, firstly, that logging records are retained for a long period, thereby ensuring continuous documentation, and secondly, that memory space is not occupied unnecessarily.

The logging records are saved by redirecting the output of FTSHWLOG (Displaying logging records, page 281) to a file, e.g. by executing the FTSHWLOG command as CLIST.

When deleting logging records, the disk storage occupied by the log file is not released. The free space within the file is, however, used to store new records. In the case of very large log files it may take several minutes to delete log records.

In this case the following procedure is recommended:

- ► Switch the log file using FTMODOPT LOGGING=*CHANGE-FILES. The current log file is switched "offline". New log records are now written to a new log file.
- ► After a certain time, evaluate all log files in the offline log file and archive them using FTSHWLOG.
- ► Delete the offline log file using FTDELLOG .

i The default setting for the command FTDELLOG has changed in openFTV11.0. If you specify the command without parameters, the default value *PARAMETERS() is used instead of *ALL as previously, i.e. all log records are deleted that have been written up to 00:00 h of the current day. This means that the command remains downward compatible in terms of its behavior.

Format

```
FTDELLOG

SELECT = *ALL / *OWN / *PARAMETERS(...) / *LOGGING-FILES (...)

   *PARAMETERS(...)
        OWNER-IDENTIFICATION = *ALL / *OWN / <name 1..8>
       ,LOGGING-DATE = *TODAY / *TOMORROW / <date 8..10>
       ,LOGGING-TIME = 00:00 / <time 1..8>
       ,RECORD-TYPE = *ALL / *PARAMETERS(...)
           *PARAMETERS(...)
                FT = *ALL / *NONE
               ,FTAC = *ALL / *NONE
               ,ADM = *ALL / *NONE
       ,LOGGING-ID = *ALL / <alphanum-name 1..12>

   *LOGGING-FILES(...)
        BEFORE = *TIME(...)
           *TIME = (...)
               DATE = <date 8..10>
              ,TIME = 00:00 / <time1..8>
```

Operands

SELECT =
Selects a group of logging records.

SELECT = *ALL
Deletes all logging records.

SELECT = *OWN
Deletes all logging records of your own ID.

SELECT = *PARAMETERS(...)

OWNER-IDENTIFICATION =
User ID whose logging records are to be deleted.

OWNER-IDENTIFICATION = *ALL
The user ID is not a selection criterion.

OWNER-IDENTIFICATION = *OWN
Logging records of the own user ID are deleted.

OWNER-IDENTIFICATION = <name 1..8>
User ID whose logging records are to be deleted.

LOGGING-DATE =
Date before which the logging records are to be deleted.

LOGGING-DATE = *TODAY
If a time was specified explicitly with LOGGING-TIME, all log records that were written before this time are deleted. If no date was specified, openFT deletes all log records that were written up to midnight inclusive of the previous day.

LOGGING-DATE = *TOMORROW
All logging records that were created before the command was input are deleted.

LOGGING-DATE = <date 8..10>
Date in the format *yyyy-mm-dd* or *yy-mm-dd*, e.g. 2016-12-24 or 16-12-24 for the 24th of December, 2016. openFT then deletes only those logging records that were written before the date and time specified with LOGGING-TIME and LOGGING-DATE.

LOGGING-TIME =
Logging records written up to the specified time are deleted.

LOGGING-TIME = 00:00
If a date was specified explicitly with LOGGING-DATE, openFT deletes all log records written before the specified date. If no date was specified, openFT deletes all log records that were written up to midnight inclusive of the previous day.

LOGGING-TIME = <time 1..8>
Time for the day specified with LOGGING-DATE. openFT deletes all log records written before this time. You specify the time in the format *hh:mm:ss*, e.g. 14:30:10.

RECORD-TYPE =
Defines the type of logging records to be deleted.

RECORD-TYPE = *ALL
The record type is not a selection criterion.

RECORD-TYPE = *PARAMETERS(...)
Type of the logging record.

> **FT = *ALL / *NONE**
> Specifies whether or not the FT logging records are to be deleted.
>
> **FTAC = *ALL / *NONE**
> Specifies whether or not FTAC logging records are to be deleted.
>
> **ADM = *ALL / *NONE**
> Specifies whether ADM log records are deleted or not.

LOGGING-ID =
Selects the logging records on the basis of the logging ID.

> **LOGGING-ID = *ALL**
> The logging ID is not a selection criterion.

LOGGING-ID = <alphanum-name 1..12>
All logging records with a logging ID smaller than or equal to the specified value are deleted.

SELECT = *LOGGING-FILES(...)
Controls the deletion of offline log files. Offline log records cannot be deleted individually: only entire files can be deleted.

BEFORE = *TIME(...)
Deletes all the offline log files which were switched offline on or before the specified time (local time!) by switching the log file offline. This ensures that only log records which are at least as old as the specified time are deleted.

If you enter the current date or a date in the future, then all the existing offline log files are deleted.

DATE = <date 8..10>
Creation date in the format *yyyy-mm-dd* or *yy-mm-dd*, e.g. 2016-03-31 or 16-03-31 for March 31, 2016.

TIME = 00:00 / <time 1..8>
Time for the date specified with DATE. You enter the time in the format *hh:mm:ss*, e.g. 14:30:10.

i Up to 1024 log files can be deleted per call. If you wish to delete more files, repeat the call.

Under some circumstances it may not be possible to immediately delete a log file which has just been switched to become an offline log file after it has been switched if the file still has synchronous requests open.

Example

The FT administrator wants to delete all existing FT log records from the current log file (If there is a large number of log records, this may take several minutes!). If FTAC is not installed, logging only contains FT log records and ADM log records where applicable. They are deleted with the following command:

```
FTDELLOG SELECT=*PARAMETERS(LOGGING-DATE=*TOMORROW)
```

The FT administrator does not need to specify the operand OWNER-IDENTIFICATION because the standard value *ALL applies.

However, if FTAC were used then this command would delete the FT and FTAC logging records and ADM logging records where applicable because both FT=*ALL and FTAC=*ALL and ADM=*ALL are default values for RECORD-TYPE. If only the FT logging records are to be deleted, but the FTAC and ADM logging records are to be retained, then the FT administrator must extend the command:

```
FTDELLOG SELECT=*PARAMETERS(LOGGING-DATE=*TOMORROW, -
                RECORD-TYPE=*PARAMETERS(FTAC=*NONE,ADM=*NONE))
```

The FT administrator wants to delete all offline log files which are set to online by switching the log file before or on June 27, 2012.

```
FTDELLOG SELECT=*LOGGING-FILES(BEFORE=*TIME(2012-06-27))
```

4.10 FTDELPRF Delete admission profile

Note on usage

User group: FTAC user and FTAC administrator

The command can be entered under TSO.

A prerequisite for using this command is the use of openFT-AC.

Functional description

With the command FTDELPRF, you can delete all admission profiles of which you are the owner. In your role as FTAC administrator, you can also delete the admission profiles of any users. 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 321), you can view the profiles and decide which ones you no longer need.

Format

```
FTDELPRF

 NAME = *ALL / <alphanum-name 1..8> / *STD
,PASSWORD = *NONE / <alphanum-name 1..8>
,SELECT-PARAMETER = *OWN / *PARAMETERS(...)
   *PARAMETERS(...)
      TRANSFER-ADMISSION = *ALL / *NOT-SPECIFIED / <alphanum-name 8..32> /
                                  <c-string 8..32 with-low> / <x-string 15..64>
     ,OWNER-IDENTIFICATION = *OWN / *ALL / <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.

The administrator can delete his/her own profiles with this entry. He/She can also use SELECT-PARAMETER to delete all the admission profiles of a particular user or all the admission profiles in the system.

NAME = <alphanum-name 1..8>
Deletes the admission profile with the specified name.

NAME = *STD
Deletes the standard admission profile for your own user ID.

PASSWORD =
You enter the FTAC password which permits you to use FTAC commands with your user ID.

PASSWORD = *NONE
No FTAC password is required.

PASSWORD = <alphanum-name 1..8>
Specifies the corresponding FTAC password as required.

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 ADMISSION.
FTAC administrators can address the admission profiles to be deleted using their TRANSFER ADMISSION or OWNER IDENTIFICATION.

SELECT-PARAMETER = *OWN
Deletes your own admission profiles.

SELECT-PARAMETER = *PARAMETERS(...)
With this structure, you can enter individual selection criteria.

TRANSFER-ADMISSION =
You can use the transfer admission of an admission profile as a selection criterion for deletion.

TRANSFER-ADMISSION = *ALL
Deletes admission profiles irrespective of the TRANSFER-ADMISSION.

TRANSFER-ADMISSION = *NOT-SPECIFIED
Deletes admission profiles for which no transfer admission is specified.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
Deletes the admission profile which is accessed with this transfer admission. The alphanumeric entry is always saved in lower-case letters. The FTAC user can only enter the transfer admissions of his/her own admission profiles.

OWNER-IDENTIFICATION =
Deletes a specific owner’s admission profile. The FTAC user can only delete his/her own profiles. The FTAC administrator can also enter foreign user IDs.

OWNER-IDENTIFICATION = *OWN
Deletes your own admission profile.

OWNER-IDENTIFICATION = *ALL
Allows the FTAC administrator to delete admission profiles of all user IDs. The FTAC user is not permitted to use this entry.

OWNER-IDENTIFICATION = <alphanum-name 1..8>
The FTAC user can only specify his/her own user ID; the effect corresponds to *OWN. The FTAC administrator deletes the admission profiles under this user ID.

Example

A user wants to delete his/her admission profile with the name *Patent*:

```
FTDELPRF PATENT
```

4.11 FTEXEC Execute remote command

Note on usage

User group: FT user

Functional description

With the FTEXEC command, you can execute operating system commands in the remote system. In the local system, the resulting standard and standard error output can be sent to *STDERR, *STDOUT or to a file.

FTEXEC is only available for openFT partners from Fujitsu Technology Solutions.

On success, FTEXEC returns 0 and if an error occurs it returns 12. The result of the command executed is also transferred.

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

In the case of output to *STDOUT, the character set specified in the local system is used.

Format

```
FTEXEC

PARTNER = <text 1..200 with-low>
,CMD= *NOT-SPECIFIED / <c-string 1..400 with-low> (...)
        |  CODED-CHARACTER-SET = *STD / <alphanum-name 1..8>
,TRANSFER-ADMISSION = *NONE / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> /
                            *PARAMETERS(...)
   *PARAMETERS(...)
        |  USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
        | ,ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
        | ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
,OUTPUT = *STDERR / *STDOUT / *FILE(...)
   *FILE(...)
        |  FILE-NAME = <filename 1..59>
        | ,CODED-CHARACTER-SET = *STD / <alphanum-name 1..8>
,DATA-TYPE = *CHARACTER / *BINARY
,DATA-ENCRYPTION = *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 “Specifying partner addresses” on page 76.

CMD =
Command in the syntax of the remote FT partner system. A command sequence in the remote system can only be processed if the remote system is using an FT product that supports this function.

CMD = *NOT-SPECIFIED
No command string is passed. *NOT-SPECIFIED must be used if an admission profile is specified in TRANSFER-ADMISSION for which a command sequence has been preset.

CMD = <c-string 1..400 with-low>
Command sequence. This command sequence may be a maximum of 400 characters in length, with special characters being counted double (as two characters).

CODED-CHARACTER-SET =
Coding (character set) to be used when reading the data from the standard output of the remote command.

CODED-CHARACTER-SET = *STD
The character set defined as standard in the remote system is used.

CODED-CHARACTER-SET = <alphanum-name 1..8>
The specified character set (CCS) is used. This must be known in the remote system. This specification must not be combined with DATA-TYPE=*BIN.

TRANSFER-ADMISSION =
Contains specifications about the transfer admission in the remote system.

TRANSFER-ADMISSION = *NONE
The remote system does not require or does not know any user admissions.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
If FTAC functionality is used in the remote system then the transfer admission for the remote system can be defined via an admission profile. In this case, only the TRANSFER-ADMISSION defined in the admission profile is used here. In the case of alphanumeric input, uppercase is converted to lowercase internally.

TRANSFER-ADMISSION = *PARAMETERS(...)
Specifies the user's identification, account number and password in the remote system. The operands in the brackets can also be used as positional operands without the associated keywords.

USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
Identification of the user in the remote system. The identification must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
Account number of the user in the remote system. The account number must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

PASSWORD =
Password allowing the user to access the remote system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
Password allowing the user to access the remote system. The password must be specified in the syntax of the remote system, must adhere to the conventions used in the remote system and must be known there.

OUTPUT =
Specifies where the data generated by the command should be output following transfer in the local system.

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

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

OUTPUT = *FILE(...)
The data is written to a file. Please note that only the data which the command specified with CMD outputs to *SYSLST (BS2000) or *STDOUT (on z/OS) or stdout (on a Unix/Windows system) is written to file.

FILE-NAME = <filename 1..59>
Name of the output file.

CODED-CHARACTER-SET =
Coding (character set) that is to be used to write the data.

CODED-CHARACTER-SET = *STD
The character set predefined by the system is used.

CODED-CHARACTER-SET = <alphanum-name 1..8>
Name of the character set (CCS) that is to be used. This character set must be known in the local system.

This specification must not be combined with DATA-TYPE=*BIN.

DATA-TYPE =
Transfer format for the data.

DATA-TYPE = *CHARACTER
The data is transferred as a text file.

DATA-TYPE = *BINARY
The data is transferred in binary form.

DATA-ENCRYPTION =
Specifies whether the data is to be transferred in encrypted form. The encryption of the request description data is not affected by this operand.

DATA-ENCRYPTION = *NO
The data is transferred unencrypted.

DATA-ENCRYPTION = *YES
The data is transferred encrypted.

Examples

1. The partner is a BS2000 system, output to the local file *ex.out*:

```
FTEXEC BS2PART,'/SH-FT-LOG ,3 ,OUTPUT=SYSLST',(userId,acct,'passw'),
OUTPUT=*FILE(ex.out),DATA-TYPE=*CHAR
```

2. The partner is a Unix system, output to *STDOUT:

```
FTEXEC PARTUX,'ftshwl -nb=10',uxtransadm,,*CHAR
```

3. The partner is a z/OS system:

```
a) FTEXEC ZOS1,'ftshwopt',transadm

b) FTEXEC ZOS2,'ftshwlog,10,out=*stdout',transadm,out=*file(ex.out)
```

4.12 FTEXPENV Export FTAC admission profiles and sets

Note on usage

User group: FTAC administrator

openFT-AC must be installed to use this command.

The command can be entered under TSO.

Functional description

The FTAC administrator can easily “move” admission profiles and sets when a user migrates from one computer to another. The commands FTEXPENV and FTIMPENV are intended for this purpose.

This commands are not available to FTAC users!

The commands only affect the currently set openFT instance. If necessary, the FTAC administrator must create them under several openFT instances.

Export files cannot be extended. They must be deleted and created again if necessary.

Format

```
FTEXPENV
--------------------------------------------------------------------------
 TO-FILE = <filename 1..46>
,USER-IDENTIFICATION = *ALL / list-poss(100): <name 1..8>
,SELECT-PARAMETER = *ALL / *PARAMETERS(...)
   *PARAMETERS(...)
     |  PROFILE-NAME = *ALL / *NONE / list-poss(100): <alphanum-name 1..8>
     | ,ADMISSION-SET = *YES / *NO
```

Operands

TO-FILE = <filename 1..46>
Name of the file in which the admission profiles and sets are output. Temporary files may not be used.

USER-IDENTIFICATION =
The user ID whose admission profiles and sets are to be output on file.

USER-IDENTIFICATION = *ALL
The admission profiles and sets of all user IDs are to be output on file.

USER-IDENTIFICATION = list-poss(100): <name 1..8>
The admission profiles and sets of the user IDs specified are to be output on file.

SELECT-PARAMETER =
Determines whether only admission profiles, only admission sets, or both are to be output on file. For admission profiles, you can select those which are to be output.

SELECT-PARAMETER = *ALL
All admission profiles and sets associated with the user ID specified under USER-IDENTIFICATION are to be output on file.

SELECT-PARAMETER = *PARAMETERS(...)
Specifies which of the admission profiles and sets associated with the USER-IDENTIFICATION are to be output on file.

PROFILE-NAME = *ALL
All admission profiles are output on file.

PROFILE-NAME = *NONE
No admission profiles are exported.

PROFILE-NAME = list-poss(100): <alphanum-name 1..8>
Only the profiles with the specified names (maximum 100) are output on file.

ADMISSION-SET = *YES
All admission sets are output on file.
Here, openFT only takes account of values that differ from the default (i.e. are not marked with a * in the FTSHWADS output). For all specifications that refer to the standard admission set, openFT takes over the current settings from the corresponding standard admission set when importing admission sets. The standard admission set itself is not output to file.

ADMISSION-SET = *NO
No admission sets are exported.

Example

The FTAC administrator wants to export all the admission profiles belonging to the user Billy to the external file BILLYPRF. The admission set is not to be exported:

```
FTEXPENV TO-FILE=BILLYPRF,USER-ID=BILLY,SEL=(PROF-NAME=*ALL,ADM-SET=*NO)
```

4.13 FTHELP Display information on reason codes in the logging records

Note on usage

User group: FT user and FT administrator

The command has to be entered in the TSO command mode.

Functional description

You can have the meaning of the reason codes contained in the logging records displayed by the command FTHELP (RC in the output of the command FTSHWLOG in logging records).

Format

FTHELP
<number 1..ffff>

Description

<number 1..ffff>
Stands for a four-digit reason code as it appears in the logging record. Leading zeros can be omitted during input. In an FTAC logging record, the reason code 0000 means that an FTAC admission check has permitted the request. Any other reason code indicates the reason for rejection by FTAC.
The reason code 0000 in an FT logging record indicates that file transfer has terminated successfully. All reason codes other than 0000 indicate failure.

Example

A transfer code is rejected by the local system with the following error message:

```
FTR2046 OPENFT: Local transfer admission invalid.
```

The FTAC administrator uses the command FTSHWLOG (see page 281) to display the relevant FTAC logging record. This is what the output he/she receives looks like:

```
TYP LOGG-ID TIME    RC    PARTNER  INITIATOR INIT USER-ADM FILENAME
2017-01-14
C        77 15:19:06 3003 >JUMBO    USER001        USER001  ABC
```

The meaning of reason code 3003 can now be determined with the command FTHELP:

```
FTHELP 3003
 3003: Request rejected. Invalid password
```

Thus, the request was rejected because an invalid password was specified.

4.14 FTIMPENV
Import FTAC admission profiles and sets

Note on usage

User group: FTAC administrator

openFT-AC must be installed to use this command.

This command can be entered under TSO.

Functional description

The FTAC administrator can easily “move” admission profiles and sets when a user migrates from one computer to another. The commands FTEXPENV and FTIMPENV are intended for this purpose. These commands cannot be used by the FTAC user.
All imported admission profiles will be first locked.
This can be seen in the FTSHWPRF command in the specification *LOCKED (by_import).
Privileged profiles lose their privileged status when imported. They will also be designated as private.

An admissions profile is otherwise only imported if its name does not exist on the destination ID.

If the target computer already has an admission profile with the same transfer admission and the admission profile is designated as private, both transfer admissions are locked. The transfer admission of the old profile is set to *DUPLICATED and the transfer admission of the imported profile is set to *NOT-SPECIFIED. If the already existing admission profile is designated as ”public”, then it is not locked.

Format

```
FTIMPENV

 FROM-FILE = <filename 1..46>
,USER-IDENTIFICATION = *ALL / list-poss(100): <name 1..8>
,SELECT-PARAMETER = *ALL / *PARAMETERS(...)
    *PARAMETERS(...)
    |    PROFILE-NAME = *ALL / *NONE / list-poss(100): <alphanum-name 1..8>
    |   ,ADMISSION-SET = *YES / *NO
,SECURITY = *STD / *HIGH
```

Operands

FROM-FILE = <filename 1..46>
Name of the file from which the admission profiles and sets are to be imported. If the file contains invalid data or if there is an error while accessing the file, the command is rejected with the message FTC0103.

USER-IDENTIFICATION =
User ID whose admission profiles and sets are to be transferred from an export file.

USER-IDENTIFICATION = *ALL
The admission profiles and sets of all users are to be transferred.

USER-IDENTIFICATION = list-poss(100): <name 1..8>
The admission profiles and sets of the users specified (maximum 100) are to be transferred.

SELECT-PARAMETER =
Determines whether only admission profiles, only admission sets, or both are to be imported. For admission profiles, you can specify which are to be imported.

SELECT-PARAMETER = *ALL
All the admission profiles and sets associated with the user ID specified under USER-IDENTIFICATION are to be imported.

SELECT-PARAMETER = *PARAMETERS(...)
Specifies which of the admission profiles and sets associated with the USER-IDENTIFICATION are to be imported.

> **PROFILE-NAME = *ALL**
> All admission profiles are to be imported.
>
> **PROFILE-NAME = *NONE**
> No admission profiles are to be imported.

PROFILE-NAME = list-poss(100): <alphanum-name 1..8>
Only the profiles specified are to be imported (maximum 100).

ADMISSION-SET = <u>*YES</u>
All admission sets are to be imported.

ADMISSION-SET = *NO
No admission sets are to be imported.

SECURITY =
An FTAC administrator with system administrator privilege can use this operand to control security.

SECURITY = <u>*STD</u>
For FTAC administrators with SU privilege:
The profile attributes are not altered when imported.

For FTAC administrators not having the SU privilege:
This operand works like the specification *HIGH, i.e. the admissions profiles are locked (locked by import) and retain the attributes USAGE=PRIVATE and PRIVILEGED = NO.

SECURITY = *HIGH
The admissions profiles are locked (locked by import) and retain the attributes USAGE=PRIVATE and PRIVILEGED=NO.

Example

The FTAC administrator wants to import all admission profiles belonging to the user Billy from the external file BILLYPRF. The admission set is not to be imported.

```
FTIMPENV FROM-FILE=BILLYPRF,USER-ID=BILLY,
         SEL=(PROF-NAME=*ALL,ADM-SET=*NO)
```

If the FTAC administrator possesses the SU privilege then the profiles can be used immediately. Otherwise, Billy must first unlock them with FTMODPRF:

```
FTMODPRF *ALL,TRANS-ADM=*OLD-ADM(VALID=*YES)
```

4.15 FTIMPKEY Import key

Note on usage

User group: FT administrator

This command can be entered under TSO.

Functional description

You can use the FTIMPKEY command as FT administrator to import a partner’s public key or an RSA key pair.

Importing a public key

If you want to import the public key of a partner, the key must have been generated by the partner’s openFT instance and the partner must have been entered in the partner list. The key is then stored in the SYSKEY file under the name of the partner. Please ensure that the partner’s instance identification is entered correctly in the partner list.

Importing an RSA key pair

You can import an RSA key pair consisting of a public and a private key. The key pair can be used for data encryption and authentication like a key pair generated by openFT.

The key pair must be generated using an external tool. It must have the length 768, 1024 or 2048 bits and be present in PEM format (openSSL native PEM or PKCS#8) or in PKCS#12 V1.0 format.

If the key pair demands a password phrase (password), then this must be specified during the import.

During import, the same applies as for key pairs generated with FTCREKEY:

- The key pair contains a unique reference number.
- The public key is stored under the name
  <openft qualifier>.<inst>.SYSPKF.R<key reference>.L<length>

For details, see section “FTCREKEY Create a key pair set” on page 117.

Format

```
FTIMPKEY

PRIVATE-KEY = *NONE / *PARAMETERS(...)

   *PARAMETERS(...)
     |  FILE-NAME = <filename 1..42>
     |  ,PASSWORD = *NONE / <c-string 1..64 with-low>
     |  ,TYPE = *PEM / *P12

,PUBLIC-KEY = *NONE / *PARAMETERS(...)

   *PARAMETERS(...)
     |  FILE-NAME = <filename 1..42>
```

Operands

PRIVATE-KEY =
Specifies whether a private key is to be imported.

PRIVATE-KEY = *NONE
No private key is imported.

PRIVATE-KEY = *PARAMETERS(...)
Defines which private key is imported.

FILE-NAME = <filename 1..42>
Name of the file which contains the private key.

PASSWORD =
Password with which the private key is protected.

PASSWORD = *NONE
The private key is not protected by a password.

PASSWORD = <c-string 1..64 with-low>
Password with which the private key is protected.

TYPE =
Type of key file whose key is to be imported.

TYPE = *PEM
The key file is available in PEM format.

TYPE = *P12
The key file contains a certificate and a private key in accordance with the standard PKCS#12 V1.0. The file is searched for a private key and any non-supported elements (e.g. certificates, CRLs) are ignored during the import. The first private key that is found in the file is imported. Any others are ignored.
If the certificate is protected by a signature or hash, then openFT does not perform a validity check. The validity of the file must be verified using other means.

PUBLIC-KEY =
Specifies whether a public key is to be imported.

PUBLIC-KEY = *NONE
No public key is imported.

PUBLIC-KEY = *PARAMETERS(...)
Defines which public key is imported.

FILE-NAME = <filename 1..42>
Name of the file which contains the public key.

ℹ You must specify a file in at least one of the operands PRIVATE-KEY or PUBLIC-KEY.

4.16 FTMGET
Fetch multiple files from a remote system

Note on usage

User group: FT user

Functional description

FTMGET allows you to fetch synchronously or asynchronously multiple files from a remote partner system. You can specify the remote files using wildcards. To do this, FTMGET uses the NCOPY (asynchronous) or the FTSCOPY (synchronous) command internally. The transfer mode (synchronous or asynchronous) is controlled via the TRANSFER-MODE operand.

Format

(part 1 of 6)

FTMGET
PARTNER = <text 1..200 with-low>
,**LOC**AL-PARAMETER = ***PAR**AMETERS (...)
***PAR**AMETERS (...)
FILE-NAME = ***SAME /** <text 2..59> / <c-string_1..512_with-low>
,PASSWORD =***NONE** / <alphanum-name 1..8>
,**TRANS**FER-**ADM**ISSION = ***SAME** / <alphanum-name 8..32> / <x-string 15..64> / <c-string 8..32 with-low> / ***PAR**AMETERS(...)
***PAR**AMETERS(...)
USER-IDENTIFICATION = <name 1..8>
,**ACCOUNT** = ***NONE** / <alphanum-name 1..40> / c-string 1..40>
,**PASS**WORD = ***NONE** / <alphanum-name 1..8>
,**CODED-CHA**RACTER-SET = ***STD** / <alphanum-name 1..8>

(part 2 of 6)

```
,REMOTE-PARAMETER = *BS2000(...) / *MSP(...) / *ANY(...)

   *BS2000(...)
        FILE-NAME = <filename 1..54> / <c-string 1..512 with-low>  / *LIBRARY-ELEMENT(...)
           *LIBRARY-ELEMENT(...)
               LIBRARY = <filename 1..54>
              ,ELEMENT = <filename 1..64>(...) / <c-string 1..64 with low>(...)
                  <filename>(...) / <c-string>(...)
                      ,VERSION = *STD / <text 1..24>
              ,TYPE = <name 1..8>
       ,PASSWORD = *SAME / *NONE / <c-string 1..4> / <x-string 1..8> /
                   <integer -2147483648..2147483647>
       ,TRANSFER-ADMISSION = <alphanum-name 8..32> / <x-string 15..64> /
                             <c-string 8..32 with-low> / *PARAMETERS(...)
          *PARAMETERS(...)
              USER-IDENTIFICATION = <alphanum-name 1..8>
             ,ACCOUNT = *NONE / <alphanum-name 1..8>
             ,PASSWORD = *NONE / <c-string 1..8> / <c-string 9..32> / <x-string 1..16>
       ,PROCESSING-ADMISSION = *SAME / *NOT-SPECIFIED / *PARAMETERS(...)
          *PARAMETERS(...)
              USER-IDENTIFICATION = <alphanum-name 1..8>
             ,ACCOUNT = *NONE / <alphanum-name 1..8>
             ,PASSWORD = *NONE / <c-string 1..8> / <c-string 9..32> / <x-string 1..16>
       ,SUCCESS-PROCESSING = *NONE / <c-string 1..1000 with-low>
       ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
       ,CODED-CHARACTER-SET = *STD / <alphanum-name 1..8>
```

(part 3 of 6)

```
*MSP(...)
  FILE-NAME = <filename 1..59> / <c-string 1..512 with-low>
 ,PASSWORD = *NONE / <alphanum-name 1..8>
 ,TRANSFER-ADMISSION = <alphanum-name 8..32>/<x-string 15..64>/<c-string 8..32 with-low> /
                                          *PARAMETERS(...)
      *PARAMETERS(...)
          USER-IDENTIFICATION = <name1..8>
         ,ACCOUNT = *NONE / <alphanum-name 1..40> / <c-string 1..40>
         ,PASSWORD = *NONE / <alphanum-name 1..8>
 ,PROCESSING-ADMISSION = *SAME / *NOT-SPECIFIED / *PARAMETERS(...)
      *PARAMETERS(...)
          USER-IDENTIFICATION = <name 1..8>
         ,ACCOUNT = *NONE / <alphanum-name 1..40> / <c-string 1..40>
         ,PASSWORD = *NONE / <alphanum-name 1..8>
 ,SUCCESS-PROCESSING = *NONE /<c-string 1..1000 with-low>
 ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
 ,CODED-CHARACTER-SET = *STD / <alphanum-name 1..8>
```

(part 4 of 6)

```
 *ANY(...)
   FILE-NAME = <c-string 1..512 with-low> / *LIBRARY-ELEMENT(...)
     *LIBRARY-ELEMENT(...)
         LIBRARY = <c-string 1..63 with-low>
        ,ELEMENT = <c-string 1..64 with-low>(...)
           <c-string 1..64 with-low>(...)
               VERSION = *NONE / *STD / <c-string 1..24 with-low>
        ,TYPE =<name 1..8>
  ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128>
  ,TRANSFER-ADMISSION = *NONE / <alphanum-name 8..32> / <x-string 15..64> /
                          <c-string 8..32 with-low> /  *PARAMETERS(...)
     *PARAMETERS(...)
        USER-IDENTIFICATION = <c-string 1..67 with-low>
        ,ACCOUNT = *NONE / <c-string 1..64 with-low>
        ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128 with-low>
  ,PROCESSING-ADMISSION = *SAME / *NONE / *PARAMETERS(...)
      *PARAMETERS(...)
        USER-IDENTIFICATION = <c-string 1..67 with-low>
        ,ACCOUNT = *NONE / <c-string 1..64 with-low>
        ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128 with-low>
  ,SUCCESS-PROCESSING = *NONE /<c-string 1..1000 with-low>
  ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
  ,CODED-CHARACTER-SET = *STD / <c-string 1..8 with-low>
```

(part 5 of 6)

```
,TRANSFER- MODE = *SYNCHRONOUS / *ASYNCHRONOUS(...)

   *ASYNCHRONOUS (…)
        ,LOCAL-PARAMETER = *PARAMETERS (...)
            *PARAMETERS (...)
                ,PROCESSING-ADMISSION = *SAME / *NOT-SPECIFIED / *PARAMETERS(...)
                    *PARAMETERS(...)
                         USER-IDENTIFICATION = <name 1..8>
                        ,ACCOUNT = *NONE / <alphanum-name 1..40> / c-string 1..40>
                        ,PASSWORD = *NONE / <alphanum-name 1..8>
                ,SUCCESS-PROCESSING = *NONE /<c-string 1..1000 with-low>
                ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
                ,LISTING = *STD / *NONE / *SYSLST / *LISTFILE / *PARAMETERS(...)
                    *PARAMETERS(...)
                        OUTPUT = *SYSLST / *LISTFILE
                        ,CONDITION = *ANY / *ON-FAILURE-ONLY
        ,PRIORITY = *NORMAL / *HIGH / *LOW
        ,START = *SOON / *EARLIEST(...)
            *EARLIEST(...)
                DATE = *TODAY / *TOMORROW / <date 8..10>
                ,TIME = 00:00 / <time 1..8>
        ,CANCEL = *NO / *AT(...)
            *AT(...)
                DATE = *TODAY / *TOMORROW / <date 8..10>
                ,TIME = 23:59 / <time 1..8>
```

(part 6 of 6)

,**CASE-SENS**ITIVE = ***BY-PARTNER** / ***N**O / *YES

,**COMP**RESS = ***NONE** / ***BYTE-REPET**ITION / ***ZIP**

,**WRITE**-MODE = ***REPLACE**-FILE / ***NEW**-FILE / ***EXT**END-FILE

,**DATA**-TYPE = ***NOT-SPECIFIED** / ***CHAR**ACTER (...) / ***BIN**ARY (...) / ***USER**

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

    TRANSPARENT = ***NO** / ***YES**

  ***BIN**ARY(...)

    TRANSPARENT = ***NO** / ***YES**

,**DATA**-**ENC**RYPTION = ***NO** / ***YES** / ***ONLY-DATA-INTEGRITY**

,**REC**ORD-**SIZE** = ***NOT-SPECIFIED** /<integer 1..32756>

,**REC**ORD-**FORM**AT = ***STD** / ***FIXED** / ***VARIABLE** / ***UNDEFINED**

,**TABULATOR** = ***AUTO** / ***ON** / ***OFF**

,**TARGET-FILE-FORM**AT = ***SAME** / ***BLOCK**-ORIENTED / ***SEQ**UENTIAL(...)

  ***SEQ**UENTIAL(…)

    RECORD-**FORM**AT = ***SAME** / ***UNDEFINED**

Operands

Only the differences compared with the FTSCOPY command (see page 255) and the NCOPY command (see page 350) are described below.

LOCAL-PARAMETER = *PARAMETERS(...)

FILE-NAME =
Determines the names of the receive files in the local system.

FILE-NAME = *SAME
The file names are taken over unchanged.

FILE-NAME = <text 2..59> / <c-string_1..512_with-low>
The file names will be prefixed.
The prefix must end with a dot (.), a slash (/) or a backslash (\). The % character is replaced by the last part of the remote file name.

The last part of the name starts after the last slash (/) or backslash (\) or a corresponding character in the remote system.

The prefix can also contain the absolute or relative path of a directory that exists on the local computer. If the specified directory does not exist, FTMGET is not executed.

Please note that the resulting file name must comply with the rules of the local system, otherwise the files will not be transferred.

REMOTE-PARAMETER = *PARAMETERS(...)

FILE-NAME = <file name with wildcard ...> /
LIBRARY-ELEMENT(ELEMENT=filename 1..64 without-gen-vers with wildcard)
Specifies which files or library elements are to be fetched from the remote system.

You can only use wildcard characters in the final part of the name following the last slash (/ with Unix systems) or backslash (\ with Windows systems). A BS2000 partner is regarded as a POSIX system if the specified file name is a POSIX pathname, i.e. starts with / or ./.

The following characters can be used to define a wildcard pattern:

* as a wildcard for any string (including an empty string).

For BS2000 partners, the following applies: If the string to be searched starts with a * (asterisk), you must double the * (i.e. **) if the searched string is followed by other characters and if you do not specify any other wildcard.

? as a wildcard for any single character.

[chars]
as a wildcard for a single character from the set specified by chars. In chars, you can list individual characters or specify one or more character ranges in the form a-z. This selects all characters a through z (inclusive).

Example: [aeiX-Z] stands for one of the characters a e i X Y Z.

\x x as a wildcard for one only of the following characters: * ? [ ] \

The backslash is used to cancel the special meaning of these characters in the specified wildcard pattern.

TRANSFER- MODE =
Specifies the transfer mode.

TRANSFER- MODE = *SYNCHRONOUS
The files are fetched synchronously. To do this openFT starts internally one FTSCOPY request for each file.

TRANSFER- MODE = *ASYNCHRONOUS(…)
The files are fetched asynchronously. To do this openFT starts internally one NCOPY request for each file. The further parameters (e.g. for follow-up processing) then apply for all transfer requests.

CASE-SENSITIVE =
Determines whether the search for the files is case sensitive or not.

CASE-SENSITIVE = *BY-PARTNER
The behavior is determined by the partner system:

- For Unix and POSIX partners, upper case and lower case is considered.
- For other partner systems, upper case and lower case is not considered.

CASE-SENSITIVE = *NO
Upper case and lower case is not considered when searching for the files.

CASE-SENSITIVE = *YES
Upper case and lower case is considered when searching for the files.

If you specify REMOTE-PARAMETER=*ANY without specifying a transfer admission with file name prefix in TRANSFER-ADMISSION, the following applies for BS2000 and z/OS partners:

- BS2000 partners: FTMGET will fetch files or library elements only if file names are given in lower case in FILE-NAME=.
- z/OS partners: FTMGET will fetch files only if file names are given in upper case in FILE-NAME=.

Result messages and return codes

On success, FTMGET issues one of the following messages:

`FTR0505 Requests carried out; <n> files were transferred` (synchronous transfer)

`FTR0511 Requests accepted; <n> file transfers initiated` (asynchronous transfer)

Where <n> stands for the number of files transferred synchronously or the number of asynchronous file transfer requests initiated. If no files that match the specified pattern were found on the remote system, the following message appears instead:

```
FTR2174 No files corresponding to specified pattern found
```

Other messages:

Output (STDERR)	Meaning
FTR0707 Invalid parameter	The syntax of the local filename or library or element or element type is invalid
FTR0865 Not all files transferred successfully	At least one source file could not be transferred to the local system.
FTR2047 Request rejected by local FTAC	Invalid values in LOCAL-PARAMETER=...
FTR2155 File / directory not found	The path for the filename given in REMOTE-PARAMETER=... is incorrect
FTR2169 Transfer admission invalid	Request rejected. Remote system: Transfer admission invalid, i.e. transfer admission incorrect or insufficient FTAC authorizations.

Example

Fetch synchronously all files from the z/OS system zos1 which start with CFG and store them in the local system with prefix SAV.

```
FTMGET ZOS1,(SAV.),(C'CFG*',,'TRANSADM')
```

The same command asynchronously and start of transfer is tomorrow:

```
FTMGET ZOS1,(SAV.),(C'CFG*',,'TRANSADM'),TRANS-MODE=*A(START=(*TOM))
```

4.17 FTMOD Modify remote file attributes

Note on usage

User group: FT user

Functional description

The FTMOD command is used to modify the attributes of a file in an FT partner system. This command does not allow directories (also no PO and PDSE datasets) to be modified.

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

openFT partners:

- File name
- Access rights

FTP partners:

- File name

Format

```
FTMOD

 PARTNER = <text 1..200 with-low>
,FILE = *NOT-SPECIFIED / <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>
,PASSWORD = *NONE / <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>
,TRANSFER-ADMISSION = *NONE / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> /
                               *PARAMETERS(...)
   *PARAMETERS(...)
     |  USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
     | ,ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
     | ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128> /<alphanum-name 1..19>
,NEW-NAME = *SAME / <filename 1..54> / <c-string 1..512 with-low>
,FILE-AVAILABILITY = *UNCHANGED
,STORAGE-ACCOUNT = *UNCHANGED
,FUTURE-FILE-SIZE = *UNCHANGED
,ACCESS-MODE = *UNCHANGED / *READ-ONLY / *READ-WRITE / *REPLACE-ALL-BY(...)
   *REPLACE-ALL-BY(...)
     |  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 = *UNCHANGED
```

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 “Specifying partner addresses” on page 76.

FILE =

Name of the file in the remote FT partner system.

FILE = *NOT-SPECIFIED
The name of the file is known to the remote system because it has already been completely defined in the addressed FTAC admission profile, for instance.

FILE = <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>
Name of the file in the remote system. It must be specified in the syntax of the remote system and conform to the conventions of the remote system.
If the file name is specified with unattached Public Volume Set, the request is rejected with the error message FTR2202.

PASSWORD =
The password that provides access to the file in the remote system. If the file in the remote system is password-protected, the password required for modifying file attributes in remote systems must be specified in these operands.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>
The password that provides access to the file in the remote system. The password must be in the syntax of the remote system and conform to the conventions of the remote system.

TRANSFER-ADMISSION =
Contains the specifications for transfer admission to the remote system for file management requests.

TRANSFER-ADMISSION = *NONE
The remote system does not require or recognize any user authorization.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
The transfer admission for the remote system can only be defined in an admission profile if the FTAC functionality is in use in the remote system. In this case, only the TRANSFER-ADMISSION defined in the FT profile is specified. The alphanumeric entry is converted internally to lowercase characters.

TRANSFER-ADMISSION = *PARAMETERS(...)
Specifies the identification, the account number and the password of the user in the remote system. The operands in brackets can also be used as positional operands without their keywords.

USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
Identification of the user in the remote system. The identification must be specified in the syntax of the remote system and must conform to the conventions of the remote system.

ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
Account number for the user in the remote system. The account number must be specified in the syntax of the remote system and must observe its conventions.

PASSWORD =
The password that allows the user to access the remote system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
The password that allows the user to access the remote system. The password must be specified in the syntax of the remote system, must conform to the conventions of the remote system, and be recognized by the remote system.

NEW-NAME =
New name of the file in the remote FT partner system.

NEW-NAME = *SAME
The previous file name remains unchanged.

NEW-NAME = <filename 1..54> / <c-string 1..512 with-low>
The new name of the file in the remote system. The previous name is no longer valid. The file name must be specified in the syntax of the remote system and conform to the conventions of the remote system.

FILE-AVAILABILITY = *UNCHANGED
The previous file availability remains unchanged.

STORAGE-ACCOUNT = *UNCHANGED
The previous account number remains unchanged.

FUTURE-FILE-SIZE = *UNCHANGED
The previous file size remains unchanged.

ACCESS-MODE =
Permitted access methods.

ACCESS-MODE = *UNCHANGED
The previous access rights remain unchanged.

ACCESS-MODE = *READ-ONLY
Short form of the current access rights READ-FILE, READ-ATTRIBUTES and CHANGE-ATTRIBUTE, and thus simplifies input.

ACCESS-MODE = *READ-WRITE
Short form of the current access rights READ-FILE, REPLACE-FILE, EXTEND-FILE, READ-ATTRIBUTES, CHANGE-ATTRIBUTES, DELETE-FILE and ERASE-DATA, and thus simplifies input.

ACCESS-MODE = *REPLACE-ALL-BY(...)
The existing access rights of the file in the remote system are replaced by the specified access rights.

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

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

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

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

CHANGE-ATTRIBUTES = <u>*NO</u> / *YES
The file attributes cannot or can be modified.

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

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

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

LEGAL-QUALIFICATION = <u>*UNCHANGED</u>
The previous legal qualifications remain unchanged.

Example

You wish to reset the access rights of the remote file MYFILE from READ-WRITE to READ-ONLY. The file is stored in the BS2000 system HUGO under the user ID JIM, with the account number A1234FT and the password C’PWD’

```
FTMOD PARTNER=HUGO,FILE-NAME=MYFILE,   -
                             TRANSFER-ADMISSION=(JIM,A1234FT,C'PWD'),-
                             ACCESS-MODE=*READ-ONLY
```

Short form:

```
FTMOD HUGO,MYFILE,,(JIM,A1234FT,'PWD'),,,,,*R-O
```

4.18 FTMODADS Modify admission set

Note on usage

User group: FTAC user and FTAC administrator

Prerequisite for using this command is the use of openFT-AC.

The command can be entered under TSO.

Functional description

The FTAC user can modify the admission set for his/her own user ID with the FTMODADS command. The FTAC administrator also can modify the admission sets of foreign user IDs. You may access two components of the admission set:

a) You can define a password to be entered for almost all subsequent FTAC commands (except the FTSHW... commands). This prevents other users working with your user ID from entering FTAC commands.

   ℹ 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.

   ⚠ **WARNING!**
   If the FTAC administrator should assign and subsequently forget a password, the FTAC environment must be reinstalled. In this case, all admission profiles and sets are deleted!

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.

   FTAC administrators assign a maximum security level for each of the six basic functions. The user ID associated with the admission set can then use this function with all partner systems with this security level or lower. The owner of the admission set may only increase the degree of restriction.

In addition, the FTAC administrator can delete an admission set from the admission file by entering the standard admission set for the user ID in question (MAX-LEVELS=*STD). This is also possible with user IDs which have already been deleted!

Format

```
FTMODADS

 USER-IDENTIFICATION = *OWN / *STD / <name 1..8>
,PASSWORD = *NONE / <alphanum-name 1..8>
,SELECT-PARAMETER = *ALL
,NEW-PASSWORD = *OLD / *NONE / <alphanum-name 1..8>
,PRIVILEGED = *UNCHANGED
,MAX-LEVELS = *UNCHANGED / *STD / <integer 0...100> / *PARAMETERS(...)
    *PARAMETERS(...)
    |    OUTBOUND-SEND = *UNCHANGED / *STD / <integer 0...100>
    |   ,OUTBOUND-RECEIVE = *UNCHANGED / *STD / <integer 0...100>
    |   ,INBOUND-SEND = *UNCHANGED / *STD / <integer 0...100>
    |   ,INBOUND-RECEIVE = *UNCHANGED / *STD / <integer 0...100>
    |   ,INBOUND-PROCESSING = *UNCHANGED / *STD / <integer 0...100>
    |   ,INBOUND-MANAGEMENT = *UNCHANGED / *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 = *STD
The standard admission set is to be modified. Only the FTAC administrator can make this entry.

USER-IDENTIFICATION = <name 1..8>
The admission set for this user ID is to be modified. The FTAC user can only enter his/her own user ID here.
The FTAC administrator can enter any user ID here.

PASSWORD =
FTAC password which authorizes you to use FTAC commands, if such a password was defined in your admission set. An FTAC password is set with the operand NEW-PASSWORD.

PASSWORD = *NONE
No FTAC password is required for this admission set.

PASSWORD = <alphanum-name 1..8>
This password authorizes this user to use FTAC commands.

SELECT-PARAMETER = *ALL
In later openFT-AC versions it will be possible to specify additional selection criteria here.

NEW-PASSWORD =
Changes the FTAC password. If such an FTAC password has already been set, it must be used for almost all FTAC commands on the user ID for this admission set (except: the FTSHW... commands). This is done using the parameter PASSWORD in the respective commands.

NEW-PASSWORD = *OLD
The FTAC password remains unchanged.

NEW-PASSWORD = *NONE
No FTAC password is required for the user ID associated with this admission set.

NEW-PASSWORD = <alphanum-name 1..8>
Specification of the new FTAC password.

PRIVILEGED = *UNCHANGED
This parameter is only supported for reasons of compatibility. Authorization of the FTAC administrator is now only possible via the FTACADM member in the openFT parameter library.

MAX-LEVELS =
You set which security level(s) you can access, with which basic functions, from the user ID of this admission set. Either you can set one security level for all basic functions or different security levels for each basic function.
The MAX-USER-LEVELS for this admission set are set by the FTAC user; the MAX-ADM-LEVELS are set by the FTAC administrator.
FTAC runs authorization checks on the basis of the lowest specified security level. FTAC users may reduce but not increase the values specified for them by the FTAC administrator, see example to FTSHWADS.

MAX-LEVELS = *UNCHANGED
The security levels set in this admission set are to remain unchanged.

MAX-LEVELS = *STD
For this admission set, the values of the standard 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 = <u>*UNCHANGED</u>
The value for OUTBOUND-SEND remains unchanged.

OUTBOUND-SEND = *STD
For OUTBOUND-SEND, the value from the standard 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 = <u>*UNCHANGED</u>
The value for OUTBOUND-RECEIVE remains unchanged.

OUTBOUND-RECEIVE = *STD
For OUTBOUND-RECEIVE, the value from the standard 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 = <u>*UNCHANGED</u>
The value for INBOUND-SEND remains unchanged.

INBOUND-SEND = *STD
For INBOUND-SEND, the value from the standard 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 standard 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 as part of their FT request.

INBOUND-PROCESSING = *UNCHANGED
The value for INBOUND-PROCESSING remains unchanged.

INBOUND-PROCESSING = *STD
For INBOUND-PROCESSING, the value from the standard 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 standard 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 set.

```
FTSHWADS
```

He receives the following output:

```
               MAX. USER LEVELS               MAX. ADM LEVELS         ATTR
USER-ID  OBS  OBR  IBS  IBR  IBP  IBF  OBS  OBR  IBS  IBR  IBP  IBF
DACKTAIL 100  100  100  100    0  100   80   80   80   80   60   60
```

Steven forbids any follow-up processing and thus only allows FT functions.

```
FTMODADS MAX-LEVELS=*PARAMETERS(INBOUND-PROCESSING = 0)
```

The short form of this command is

```
FTMODADS MAX-LEV=(IN-PROC=0)
```

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

```
FTSHWADS
```

He receives the following output:

```
               MAX. USER LEVELS                MAX. ADM LEVELS        ATTR
 USER-ID  OBS  OBR  IBS  IBR  IBP  IBF    OBS  OBR  IBS  IBR  IBP  IBF
 DACKTAIL 100  100  100  100    0  100     80   80   80   80   60   60
```

Although the FTAC administrator permitted follow-up processing (IBP) for all partners with a security level of 60 or lower, this is no longer possible on Steven's user ID. However, Steven then sets up a profile for trustworthy partners which allows them follow-up processing again.

```
FTCREPRF FRIENDS,TRANS-AD=‘for my friends‘,IGN-MAX-LEV=(IN-PROC=*YES), -
      PROCESSING-ADMISSION=(STEVEN,XXXX,PASSWORD)
```

Example

Jack John, the FTAC administrator of the Dack Bank, wishes set up the admission set for his employee Steven, such that Steven

– can send files to partner systems with the security level of 10 or lower (basic function “outbound send”),

– can request files from partner systems with the security level of 10 or lower (basic function “outbound receive”).

He wants all partner systems to be able send files to and request files from the user ID STEVEN. Therefore he sets the security level for INBOUND-SEND and INBOUND-RECEIVE to 100.

Jack does not wish to permit follow-up processing to be initiated from external partners, since he is too stingy to want to make his resources available to others. Therefore, he sets INBOUND-PROCESSING and INBOUND-MANAGEMENT at 0. Since these values are set in the standard admission set for the Dack Bank, these specifications are used for *STD. No FTAC password is defined.

The long form of the required command is as follows:

```
FTMODADS USER-IDENTIFICATION=STEVEN,    -
         MAX-LEVELS=(OUTBOUND-SEND=10,  -
         OUTBOUND-RECEIVE=10,           -
         INBOUND-SEND=100,              -
         INBOUND-RECEIVE=100,           -
         INBOUND-PROCESSING=*STD,       -
         INBOUND-MANAGEMENT=*STD)
```

A possible short form of this command would be:

```
FTMODADS STEVEN,MAX-LEV=(10,10,100,100,*STD,*STD)
```

4.19 FTMODDIR Modify remote directory attributes

Note on usage

User group: FT user

Functional description

With the FTMODDIR command, you can modify the attributes of a directory in an FT partner system. It is currently only possible to change the directory name.

Format

```
FTMODDIR

 PARTNER = <text 1..200 with-low>
,DIRECTORY-NAME = *NOT-SPECIFIED / <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>
,PASSWORD = *NONE / <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>
,TRANSFER-ADMISSION = *NONE / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> /
                             *PARAMETERS(...)
   *PARAMETERS(...)
   |    USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
   |   ,ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
   |   ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
,NEW-NAME = *SAME / <filename 1..54> / <c-string 1..512 with-low>
```

Operands

PARTNER = <text 1..200 with-low>
Name of the partner system as defined in the partner list by the FT administrator or the partner system address. For more information on address specifications, see section “Specifying partner addresses” on page 76.

DIRECTORY-NAME =
Name of the directory in the remote FT partner system.

DIRECTORY-NAME = *NOT-SPECIFIED
The name of the directory is known to the remote system because it has already been completely defined in the addressed FTAC admission profile, for instance.

DIRECTORY-NAME = <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>
Name of the directory in the remote system. This must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

PASSWORD =
Password permitting unrestricted access to the directory in the remote system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>
Password making it possible to access the directory in the remote system. The password must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

TRANSFER-ADMISSION =
Contains specifications concerning the transfer admission in the remote system for the file management request.

TRANSFER-ADMISSION = *NONE
The remote system does not require or does not know any user admissions.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
If FTAC functionality is used in the remote system then the transfer admission for the remote system can be defined via an admission profile. In this case, only the TRANSFER-ADMISSION defined in the admission profile is used here. The alphanumeric input is converted to lowercase internally.

TRANSFER-ADMISSION = *PARAMETERS(...)
Specifies the user's identification, account number and password in the remote system. The operands in the brackets can also be used as positional operands without the associated keywords.

> **USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>**
> Identification of the user in the remote system. The identification must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.
>
> **ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>**
> Account number of the user in the remote system. The account number must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.
>
> **PASSWORD =**
> Password allowing the user to access the remote system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
Password allowing the user to access the remote system. The password must be specified in the syntax of the remote system, must adhere to the conventions used in the remote system and must be known there.

NEW-NAME =
New name of the directory in the remote FT partner system.

NEW-NAME = *SAME
The directory name is unchanged.

NEW-NAME = <filename 1..54> / <c-string 1..512 with-low>
New name of the directory in the remote system. The previous directory name becomes invalid. The directory name must be specified in the syntax of the remote system and must adhere to the conventions used in the remote system.

Example

You wish to rename the PO library OTTO.CLIST to HUGO.CLIST on a remote z/OS system with the partner name ZOS2PART. The library is located under the account OPFT000:

```
FTMODDIR ZOS2PART,DIR-NAME='OPFT000.OTTO.CLIST'
        ,TRANS-ADM=(OPFT000,ACCT,PASSWORD)
        ,NEW-NAME=HUGO.CLIST
```

4.20 FTMODKEY Modify key

Note on usage

User group: FT administrator

Functional description

You can use the FTMODKEY command to modify the expiration date and authentication level of keys that are used for the authentication of partner systems. The changes are stored in the relevant key file.

Once the expiration date of a key has been reached, authentication using this key is rejected. However, you can still modify the expiration date after the key's validity date has expired, e.g. in order to temporarily re-enable a key so that a current key can be transferred securely.

Format

FTMODKEY
PARTNER-**NAME** = ***ALL** / <name 1..8>
,**AUTH**ENTICATION-**LEV**EL = ***UNCHA**NGED / <integer 1..2>
,**EXP**IRATION-**DATE** = ***UNCHA**NGED / *NONE / <date 8..10>

Operands

PARTNER-NAME =
Specifies the partner whose key is to be modified.

PARTNER-NAME = *ALL
The installed keys of all partner systems are modified.

PARTNER-NAME = <name 1..8>
Name of the partner whose key is modified.

AUTHENTICATION-LEVEL =
Specifies the authentication level for the key or keys.

AUTHENTICATION-LEVEL = *UNCHANGED
The authentication level remains unchanged.

AUTHENTICATION-LEVEL = 1
The authentication level for the partner or partners is set to 1. This corresponds to the options available up to openFT V11.0A.

If the partner system is subsequently authenticated at level 2, then the entry AUTHENTICATION-LEVEL=2 is automatically recorded in its key file.

AUTHENTICATION-LEVEL = 2
The partner system supports the level 2 authentication procedure introduced in openFT V11.0B . Level 1 authentication attempts are rejected.

EXPIRATION-DATE =
Specifies the expiration date of the key or keys.

EXPIRATION-DATE = <u>*UNCHANGED</u>
The expiration date remains unchanged.

EXPIRATION-DATE = *NONE
No expiration date for the key or keys.

EXPIRATION-DATE = <date 8..10>
Expiration date in the format *yyyy-mm-dd* or *yy-mm-dd*, e.g.. 2017-12-31 or 17-12-31 for December 31, 2017. The key or keys can be used for authentication at the latest up until the time 00:00 on the specified date.

4.21 FTMODOPT Modify operating parameters

Note on usage

User group: FT administrator

Functional description

The FTMODOPT command is used to modify one or more operating parameters of the local system. The relationships between the different operating parameters are explained in manual "openFT (z/OS) - Installation and Operation".

The FTMODOPT command also enables you to do the following:

- Activate and deactivate the FT trace function and console and ADM traps
- Control FT logging, monitoring and user data encryption

i Any unspecified operating parameters remain unchanged. The current operating parameters can be queried at any time using the FTSHWOPT command (see page 313).

Format

(part 1 of 3)

```
FTMODOPT

 PROCESS-LIMIT = *UNCHANGED / <integer 1..32>

,CONNECTION-LIMIT = *UNCHANGED / <integer 1..99>

,REQUEST-WAIT-LEVEL = *UNCHANGED

,PACING = *UNCHANGED

,TRANSPORT-UNIT-SIZE = *UNCHANGED / <integer 512..65535>

,SECURITY-LEVEL = *UNCHANGED / *BY-PARTNER-ATTRIBUTES / <integer 1..100>

,PARTNER-CHECK = *UNCHANGED / *STD / *TRANSPORT-ADDRESS

,TRACE = *UNCHANGED / *ON / *OFF / *CHANGE-FILES / *PARAMETERS(...)

    *PARAMETERS(...)
      |   SWITCH = *UNCHANGED / *ON / *OFF / *CHANGE-FILES
      |  ,PARTNER-SELECTION = *UNCHANGED / *ALL / *NONE / list-poss(3): *OPENFT / *FTP / *ADM
      |  ,REQUEST-SELECTION = *UNCHANGED / *ALL / list-poss(2): *ONLY-SYNC / *ONLY-ASYNC /
      |                                             *ONLY-LOCAL / *ONLY-REMOTE
      |  ,OPTIONS = *UNCHANGED / *NONE / list-poss(1): *NO-BULK-DATA

,LOGGING = *UNCHANGED / *CHANGE-FILES / *SELECT(...)

    *SELECT(...)
      |   TRANSFER-FILE = *UNCHANGED / *OFF / *ON / *FAILURE
      |  ,FTAC = *UNCHANGED / *ON / *REJECTED / *MODIFICATIONS
      |  ,ADM = *UNCHANGED / *OFF / *ON / *FAILURE/ *MODIFICATIONS

,MAX-INBOUND-REQUEST = *UNCHANGED

,REQUEST-LIMIT = *UNCHANGED / <integer 2..32000>

,MAX-REQUEST-LIFETIME = *UNCHANGED / *UNLIMITED / <integer 1..400>

,SNMP-TRAPS = *UNCHANGED / *NONE
      |
```

(part 2 of 3)

```
,CONSOLE-TRAPS =  *UNCHANGED / *ALL / *NONE / *PARAMETERS(...)

   *PARAMETERS(...)
        SUBSYSTEM-STATE = *UNCHANGED / *OFF / *ON
       ,FT-STATE = *UNCHANGED / *OFF / *ON
       ,PARTNER-STATE = *UNCHANGED / *OFF / *ON
       ,PARTNER-UNREACHABLE = *UNCHANGED / *OFF / *ON
       ,REQUEST-QUEUE-STATE = *UNCHANGED / *OFF / *ON
       ,TRANSFER-SUCCESS = *UNCHANGED / *OFF / *ON
       ,TRANSFER-FAILURE = *UNCHANGED / *OFF / *ON

,HOST-NAME = *UNCHANGED

,IDENTIFICATION = *UNCHANGED / <c-string 1..64 with-low> / <composed-name 1..64>

,KEY-LENGTH = *PARAMETERS (...)

   *PARAMETERS(...)
       RSA-PROPOSED = *UNCHANGED / 0 / 768 / 1024 / 2048
       ,RSA-MINIMUM =  *UNCHANGED / 0 / 768 / 1024 / 2048
       ,AES-MINIMUM =  *UNCHANGED / *NONE / 128 / 256

,CODED-CHARACTER-SET = *UNCHANGED / <alphanum-name 1..8>

,OPENFT-APPLICATION = *UNCHANGED / *STD / <text 1..24>

,OPENFT-STD = *UNCHANGED / *STD /  <integer 1..65535>

,FTAM-APPLICATION = *UNCHANGED

,FTP-PORT = *UNCHANGED / *NONE / *STD / <integer 1..65535>

,DYNAMIC-PARTNERS = *UNCHANGED / *OFF / *ON

,ADM-PORT = *UNCHANGED / *STD / <integer 1..65535>

,ACTIVE-APPLICATIONS = *UNCHANGED / *ALL / *NONE / list-poss(3): *OPENFT / *ADM / *FTP

,ADM-CONNECTION-LIMIT = *UNCHANGED / <integer 1..99>

,MONITORING  = *UNCHANGED / *ON / *OFF / *PARAMETERS(...)

   *PARAMETERS(...)
       SWITCH = *UNCHANGED / *ON / *OFF
       ,PARTNER-SELECTION = *UNCHANGED / *ALL / *NONE / list-poss(2): *OPENFT / *FTP
       ,REQUEST-SELECTION = *UNCHANGED / *ALL / list-poss(2): *ONLY-SYNC / *ONLY-ASYNC /
                                         *ONLY-LOCAL / *ONLY-REMOTE
```

(part 3 of 3)

```
,ADM-TRAPS = *UNCHANGED / *NONE / *PARAMETERS(...)

   *PARAMETERS(...)
    |   DESTINATION = *UNCHANGED / *NONE / *PARAMETERS(...)
    |      *PARAMETERS(...)
    |       |   PARTNER = *UNCHANGED / <text 1..200 with-low>
    |       |   ,TRANSFER-ADMISSION = *UNCHANGED / <alphanum-name 8..32> /
    |       |                                <c-string 8..32 with-low> / <x-string15..64> /
    |   ,SELECTION = *UNCHANGED / *ALL/ *NONE / *PARAMETERS(...)
    |      *PARAMETERS(...)
    |       |   ,FT-STATE = *UNCHANGED / *OFF / *ON
    |       |   ,FT-STATE = *UNCHANGED / *OFF / *ON
    |       |   ,PARTNER-STATE = *UNCHANGED / *OFF / *ON
    |       |   ,PARTNER-UNREACHABLE = *UNCHANGED / *OFF / *ON
    |       |   ,REQUEST-QUEUE-STATE = *UNCHANGED / *OFF / *ON
    |       |   ,TRANSFER-SUCCESS = *UNCHANGED / *OFF / *ON
    |       |   ,TRANSFER-FAILURE = *UNCHANGED / *OFF / *ON

,ENCRYPTION-MANDATORY = *UNCHANGED / *NO / list-poss(2): *INBOUND / *OUTBOUND

,DELETE-LOGGING = *UNCHANGED / *PARAMETERS(...)

   *PARAMETERS(...)
    |   SWITCH = *UNCHANGED / *ON / *OFF
    |   ,RETENTION-PERIOD = *UNCHANGED / <integer 0..999 days>
    |   ,REPEAT = *UNCHANGED / *DAILY / *WEEKLY(...) / *MONTHLY(...)
    |      *WEEKLY(...)
    |       |   ON = *SUNDAY / *MONDAY / *TUESDAY / *WEDNESDAY / THURSDAY / *FRIDAY /
    |       |          *SATURDAY
    |      *MONTHLY(...)
    |       |   ON = 1 / <integer 1..31>
    |   ,DELETE-TIME = *UNCHANGED / <time 1..8>

,RECOVERY-OUTBOUND = *UNCHANGED / *ON / *OFF

,RECOVERY-INBOUND = *UNCHANGED / *ON / *OFF
```

Operands

PROCESS-LIMIT =

Maximum number of tasks that can be reserved simultaneously for the execution of file transfer requests.

Default setting following installation: 2

PROCESS-LIMIT = *UNCHANGED

PROCESS-LIMIT is not changed, default value.

PROCESS-LIMIT = <integer 1..32>
PROCESS-LIMIT can have any value between 1 and 32.

CONNECTION-LIMIT =
Maximum number of transport connections that can be reserved for the execution of FT requests. This limit does not include file management requests and synchronous requests. The maximum number of transport connections cannot be exceeded, not even if there are many high-priority file transfer requests to be executed. Since only one request can be processed at a time per transport connection, CONNECTION-LIMIT is also the maximum number of requests which a system can process simultaneously. One third of the transport connections defined by CONNECTION-LIMIT are reserved for requests from the remote system, and another third for requests submitted in the local system. The remaining third are available for both local and remote requests. This prevents locally submitted requests from blocking the system against requests from remote systems. If CONNECTION-LIMIT is less than 3, no transport connections are reserved.

Default setting following installation:16

CONNECTION-LIMIT = *UNCHANGED
The CONNECTION-LIMIT value is not changed, default value.

CONNECTION-LIMIT = <integer 1..99>
CONNECTION-LIMIT can have any value between 1 and 99.

REQUEST-WAIT-LEVEL = *UNCHANGED
The value for REQUEST-WAIT-LEVEL is unchanged.

PACING = *UNCHANGED
This parameter is only supported for reasons of compatibility an cannot be modified.

TRANSPORT-UNIT-SIZE =
Maximum size of a transport unit in bytes.

Default setting following installation: 65535 bytes

TRANSPORT-UNIT-SIZE = *UNCHANGED
The current value size of a transport unit in bytes is unchanged.

TRANSPORT-UNIT-SIZE = <integer 512..65535>
TRANSPORT-UNIT-SIZE can assume any value between 512 and 65535.
It is recommended that you use value 65535.

SECURITY-LEVEL =
This parameter need only be specified when FTAC functionality is used. An important part of the access protection functions provided by this product is based on the allocation of a security level to each partner. These security levels are designated using integers. The FT administrator can define a global value. This security level applies to all partner systems in the partner list that are not explicitly assigned their own security levels when entered.

Default setting following installation: *BY-PARTNER-ATTRIBUTES

SECURITY-LEVEL = _*UNCHANGED_
The security level is unchanged.

SECURITY-LEVEL = *BY-PARTNER-ATTRIBUTES
If you set the operand to *BY-PARTNER-ATTRIBUTES then the security level is defined automatically. This setting assigns partners that are authenticated by openFT the security level 10. Partners that are known in the transport system are assigned the security level 90. All other partners are assigned security level 100.

SECURITY-LEVEL = <integer 1..100>
SECURITY-LEVEL can assume any value between 1 and 100. If FTAC functionality is to be used, remember that 1 is the lowest level of security, offering the least protection. This is sufficient if you do not wish to further differentiate your remote systems; otherwise, a higher value should be defined. The allocation of different security levels is particularly meaningful if the authentication check is activated.

PARTNER-CHECK =
Activates the extended authentication check. When using expanded sender checking, not only the partner identification is checked, but also the transport address. PARTNER-CHECK only affects named openFT partners that are not authenticated in the in the current openFT instance (see openFT manual "Concepts and Functions").

The globally set expanded sender checking can be modified for specific partners, see the operand PARTNER-CHECK for the FTADDPTN and FTMODPTN commands.

Default setting following installation: *STD

PARTNER-CHECK = _*UNCHANGED_
The existing value is retained.

PARTNER-CHECK = *STD
If dynamic partners are prohibited (DYNAMIC-PARTNERS=*OFF), a check is performed to determine whether the partner is entered in the partner list as a partner system with his/her instance identification, and only then will the file transfer be allowed.
If dynamic partners are permitted (DYNAMIC-PARTNERS=*ON), transfers are also permitted from partners that are accessed only using their address or are not entered in the partner list at all.

PARTNER-CHECK = *TRANSPORT-ADDRESS
Extended authentication check. In addition to checking whether the partner is entered in its own partner list as a partner system, it is checked whether the transport address under which the partner logs on matches the transport address entered in the partner list for the partner system. In the FTSHWOPT command then PARTNER-CHECK = ADDR is output. This setting has no significance for dynamic partners and FTP partners.

TRACE =
Defines the settings for the FT trace functions.

Default setting following installation: *OFF

TRACE = <u>*UNCHANGED</u>
The existing FT trace functions remain unchanged.

TRACE = *ON
Switches the FT trace functions on. If the trace function is already switched on, the command FTMODOPT TRACE=*ON(...) has no effect; the trace scope cannot be modified for a trace run that is already underway.

TRACE = *OFF
Switches the FT trace functions off.

TRACE = *CHANGE-FILES
Switches to a new trace file. This allows a continuous trace to be created across several files to prevent a single trace file from becoming too large.

TRACE = *PARAMETERS(...)
Option that is to be applied when writing the trace.

SWITCH =
Deactivates the FT trace functions for the selected partners.
Default setting following installation: *OFF

SWITCH = <u>*UNCHANGED</u>
The previous value is unchanged.

SWITCH = *ON
Activates the FT trace functions.

SWITCH = *OFF
Deactivates the FT trace functions.

SWITCH = *CHANGE-FILES
Switches to a new trace file. This allows a continuous trace to be created across several files to prevent a single trace file from becoming too large.

PARTNER-SELECTION =
Selects the partners that are to be traced. The selection made here can be modified with the TRACE operand of the FTMODPTN command.
Default setting following installation: *ALL

PARTNER-SELECTION = <u>*UNCHANGED</u>
The previous value is unchanged.

PARTNER-SELECTION = *ALL
All the partners are selected for tracing.

PARTNER-SELECTION = *NONE
No partner is selected for tracing. Only those partners are traced which have been selected for tracing with the TRACE operand of the FTMODPTN command.

PARTNER-SELECTION = *OPENFT
All partners which are addressed via the openFT protocol are selected for tracing.

PARTNER-SELECTION = *FTP
All partners which are addressed via the FTP protocol are selected for tracing.

PARTNER-SELECTION = *ADM
All administration partners are selected for tracing.

REQUEST-SELECTION =
Selects the request types that are to be traced.
Default setting following installation: *ALL

REQUEST-SELECTION = *UNCHANGED
The previous value is unchanged.

REQUEST-SELECTION = *ALL
All the requests are selected for tracing.

REQUEST-SELECTION = *ONLY-SYNC
All synchronous requests are selected for tracing. Synchronous requests are always issued locally.

REQUEST-SELECTION = *ONLY-ASYNC
All asynchronous requests are selected for tracing. Requests issued remotely are always regarded as asynchronous.

REQUEST-SELECTION = *ONLY-LOCAL
All locally submitted requests are selected for tracing.

REQUEST-SELECTION = *ONLY-REMOTE
All remotely submitted requests are selected for tracing.

OPTIONS =
Controls the options for the trace functions.
Default setting following installation: *NONE

OPTIONS = *UNCHANGED
The previous value is unchanged.

OPTIONS = *NONE
No options are selected for the trace functions.

OPTIONS = *NO-BULK-DATA
If file contents (bulk data) are transferred with a protocol element and multiple trace records with the same protocol element occur in succession then only the first of these records is written to the trace file. This reduces the volume of the trace file.

LOGGING =
Switches the logging functions.

LOGGING = *UNCHANGED
The existing logging functions remain unchanged.

LOGGING = *CHANGE-FILES
The log file is changed.
The new log file is created under the name SYSLOG.Lyymmdd.Lhhmmss. *yymmdd* is the date (year, month, day) and *hhmmss* is the time (hour, minute, second in GMT) on/at which the file was created.
The old log file is closed and remains stored as an offline log file.
If the timestamp of the log file name is truncated because of the length of the openFT qualifier and/or because of the length of the LOGFILE_2ND_Q parameter (see the manual "openFT (z/OS) - Installation and Operation"), the log file can be changed only once per minute, per hour or per day. If there is no timestamp, the log file cannot be changed.

LOGGING = *SELECT(...)
Controls logging for FT, FTAC and administration functions.
Default setting following installation: *ON for all types of log records

TRANSFER-FILE = *UNCHANGED
The previous settings for FT logging remain unchanged.

TRANSFER-FILE = *OFF
Switches the FT logging functions off.

TRANSFER-FILE = *ON
Switches the FT logging functions on.

TRANSFER-FILE = *FAILURE
Only failed requests are written to the logging file.

FTAC = *UNCHANGED
The previous settings for FTAC logging remain unchanged.

FTAC = *ON
Switches the FTAC logging functions on.

FTAC = *REJECTED
All requests rejected by FTAC are logged.

FTAC = *MODIFICATIONS
All modifying requests are logged.

ADM = *UNCHANGED
The previous settings for administration logging remain unchanged.

ADM = *OFF
Deactivates administration logging.

ADM = *ON
Activates administration logging.

ADM = *FAILURE
Only failed administration requests are written to the log file.

ADM = *MODIFICATIONS
Only administration requests that modify data are written to the log file.

MAX-INBOUND-REQUEST = <u>*UNCHANGED</u>
MAX-INBOUND-REQUEST is now only supported for reasons of compatibility.

REQUEST-LIMIT =
Changes the number of requests which can be saved in the request queue.
It generally makes no sense to reduce the size of the request queue. If you increase the size, this only takes effect after openFT has subsequently been stopped, the batch job (or started task) has been terminated and then the batch job (or started task) and openFT have been restarted.

Default setting following installation: 2000

REQUEST-LIMIT = <u>*UNCHANGED</u>
The previous value remains unchanged.

REQUEST-LIMIT = <integer 2..32000>
The maximum number of requests which can be saved in the request queue is changed to the value specified.

MAX-REQUEST-LIFETIME =
Limits the lifetime of FT requests in the request file. The maximum lifetime applies to inbound and outbound requests and is specified in days. The default value when a new request file is generated is 30 days.
This parameter also affects requests to which a cancel time was explicitly assigned on request allocation (in z/OS, with the CANCEL parameter in the NCOPY command). The request is aborted as soon as either the lifetime expires or the cancel time is reached, depending on which occurs first.

Default setting following installation: 30 days

MAX-REQUEST-LIFETIME = <u>*UNCHANGED</u>
The previous value remains unchanged.

MAX-REQUEST-LIFETIME = *UNLIMITED
The lifetime of FT requests is unlimited.

MAX-REQUEST-LIFETIME = <integer 1..400>
The maximum lifetime for FT requests may have a value of between 1 and 400 days.

SNMP-TRAPS = <u>*UNCHANGED</u> / *NONE
SNMP traps are not supported by openFT in z/OS systems. The parameter is only present to ensure compatibility with other openFT products and should not be changed.

CONSOLE-TRAPS =
Activates or deactivates console traps.
By default, these trap messages are logged as asynchronous messages.
They can therefore cause storage problems on systems with high request volumes.
By default, the output of console traps is activated.
Default setting following installation: *OFF

CONSOLE-TRAPS = *UNCHANGED
The previous value is unchanged.

CONSOLE-TRAPS = *ALL
openFT outputs the FTR03XX console messages as asynchronous messages. Like the other asynchronous messages, the console messages are written to the job log. Depending on the ROUTCDE parameter in the PARM member of the parameter library, these messages may also optionally be written to a console.

CONSOLE-TRAPS = *NONE
The FTR03XX console messages are not output.

CONSOLE-TRAPS = *PARAMETERS(...)
Explicit specification of the events for which FTR03XX console messages are output.

SUBSYSTEM-STATE =
Controls the output of console messages concerning the status of the openFT subsystems.
Default setting following installation: *OFF

SUBSYSTEM-STATE = *UNCHANGED
The previous value is unchanged.

SUBSYSTEM-STATE = *OFF
No console messages concerning the status of the openFT subsystem are output.

SUBSYSTEM-STATE = *ON
Console messages concerning the status of the openFT subsystem are output.

FT-STATE =
Controls the output of console messages concerning the status of the openFT control process.
Default setting following installation: *OFF

FT-STATE = *UNCHANGED
The previous value is unchanged.

FT-STATE = *OFF
No console messages concerning the status of the openFT control process are output.

FT-STATE = *ON
Console messages concerning the status of the openFT control process are output.

PARTNER-STATE =
Controls the output of console messages concerning the status of the partner systems.
Default setting following installation: *OFF

PARTNER-STATE = *UNCHANGED
The previous value is unchanged.

PARTNER-STATE = *OFF
No console messages concerning the status of partner systems are output.

PARTNER-STATE = *ON
Console messages concerning the status of partner systems are output.

PARTNER-UNREACHABLE =
Controls the output of console messages if partner systems cannot be accessed.
Default setting following installation: *OFF

PARTNER-UNREACHABLE = *UNCHANGED
The previous value is unchanged.

PARTNER-UNREACHABLE = *OFF
No console messages are output if partner systems cannot be accessed.

PARTNER-UNREACHABLE = *ON
Console messages are output if partner systems cannot be accessed.

REQUEST-QUEUE-STATE =
Controls the output of console messages concerning the status of the request queue.
Default setting following installation: *OFF

REQUEST-QUEUE-STATE = *UNCHANGED
The previous value is unchanged.

REQUEST-QUEUE-STATE = *OFF
No console messages concerning the status of the request queue are output.

REQUEST-QUEUE-STATE = *ON
Console messages concerning the status of the request queue are output.

TRANSFER-SUCCESS =
Controls the output of console messages when a request is terminated successfully.
Default setting following installation: *OFF

TRANSFER-SUCCESS = *UNCHANGED
The previous value is unchanged.

TRANSFER-SUCCESS = *OFF
No console messages are output if a request is terminated successfully.

TRANSFER-SUCCESS = *ON
Console messages are output if a request is terminated successfully.

TRANSFER-FAILURE =
Controls the output of console messages when a request fails.
Default setting following installation: *OFF

TRANSFER-FAILURE = *UNCHANGED
The previous value is unchanged.

TRANSFER-FAILURE = *OFF
No console messages are output if a request fails.

TRANSFER-FAILURE = *ON
Console messages are output if a request fails.

HOST-NAME = *UNCHANGED
The host name remains unchanged. This parameter is supported for reasons of compatibility only.

IDENTIFICATION =
Local instance ID of your openFT instance. With the aid of this instance ID, openFT partners as of V8.1 manage the resources for your openFT instance.

The instance ID must be unique, network-wide and must not be case-sensitive. An instance ID may consist of alphanumeric characters or special characters and may have a maximum length of 64 characters. It is advisable only to use the special characters ".", "-", ":" or "%". The initial character must be alphanumeric or the special character "%". The character "%" may only be used as an initial character. The character "." must be followed by an alphanumeric character. For further details on assigning instance identifications, see section "Instance identifications" on page 88.

Default setting following installation: When an instance is installed for the first time, the VTAM name of the real host under which their instance operates is entered as the default setting. If another identification is to be used for operation then this must be configured with FTMODOPT.

IDENTIFICATION = *UNCHANGED
The instance ID remains unchanged.

IDENTIFICATION = <c-string 1..64 with-low> / <composed-name 1..64>
The instance ID is set to this value.

KEY-LENGTH = *PARAMETERS(...)
Configuration of the length of the RSA key and the length of the AES key.

RSA-PROPOSED
Length of the RSA key used for encryption. This key is used only to encrypt the AES key which is agreed on between the partners. openFT uses the AES key to encrypt the request description data and possibly also the file contents.
Default setting following installation: 2048

RSA-PROPOSED = 0
Encryption is switched off.

RSA-PROPOSED = 768 / 1024 / 2048
Length of the RSA key (in bits) that is used for the transfer of the AES key of the session.

RSA-MINIMUM
Minimum length of the RSA key (in bits) accepted by the partner system for the transfer of the AES key for the request initiated by the partner.
Default setting following installation: 0

RSA-MINIMUM = 0
There is no configuration for the minimum length of the RSA key. Every key length is accepted and even requests without encryption can be processed.

RSA-MINIMUM = 768 / 1024 / 2048
Keys with this minimum length are accepted only. If the initiator uses a shorter key he gets a counter proposal of the responder of the session. A session without encryption will be denied.

AES-MINIMUM
Minimum length of the AES key (in bits) that will be required from the partner system.
Default setting following installation: *NONE

AES-MINIMUM = *NONE
There is no configuration for the minimum length of the AES key. Every AES key length and even requests with DES keys are accepted.

AES-MINIMUM = 128 / 256
AES keys with this minimum length are accepted only. If the partner cannot fulfill this requirement the request will be rejected.

CODED-CHARACTER-SET =
Coding (character set) to be used when reading or writing a local text file during a transfer request. You can explicitly assign a different character set to a file in the transfer request or using the FT parameter library (see manual "openFT (z/OS) - Installation and Operation").
Default setting following installation: IBM1047

CODED-CHARACTER-SET = *UNCHANGED
The character set used by default to read or write the local file is the character set that is set in the system.

CODED-CHARACTER-SET = <alphanum-name 1..8>
Name of the character set used by default to read or write a local text file. The character set must be known in the local system. openFT provides a range of character sets.
This specification is only relevant for requests to openFT partners.

OPENFT-APPLICATION =
Specifies a port number and/or a transport selector for the local openFT server. Use this function carefully as it will be more difficult for the openFT partners to address the local system if the port number or transport selector differ from the default values!
Default setting following installation: *STD

OPENFT-APPLICATION = <u>*UNCHANGED</u>
The previous value is unchanged.

OPENFT-APPLICATION = *STD
The port number and transport selector are set to the default value, i.e.:
Port number: 1100
Transport selector: $FJAM in EBCDIC code, followed by three spaces.

OPENFT-APPLICATION = <text 1..24>
Valid port number and/or a transport selector in the form [<port number>].[tsel].

OPENFT-STD =
Port number other than the default when addressing openFT partners via their host names. Use this function carefully, as changing the port number from the default value means that it will no longer be possible to reach openFT partners which use the default port number and are addressed via the host name!
Default setting following installation: *STD

OPENFT-STD = <u>*UNCHANGED</u>
The previous value is unchanged.

OPENFT-STD = *STD
The port number is set to the default value 1100.

OPENFT-STD = <integer 1..65535>
Valid port number.

FTAM-APPLICATION = <u>*UNCHANGED</u>
This value is not relevant for z/OS systems and cannot be changed.

FTP-PORT =
This option allows you to specify the port number used by FTP.
Default setting following installation: 21

FTP-PORT = <u>*UNCHANGED</u>
The previous value is unchanged.

FTP-PORT = *NONE
This setting is only supported for reasons of compatibility.
You should use the operand ACTIVE-APPLICATIONS to activate and deactivate the inbound FTP server.

FTP-PORT = *STD
The port number is set to the default value 21.

FTP-PORT = <integer 1..65535>
Valid port number.

DYNAMIC-PARTNERS =
Specifies whether dynamic partners are permitted.
Default setting following installation: *ON

DYNAMIC-PARTNERS = <u>*UNCHANGED</u>
The previous value is unchanged.

DYNAMIC-PARTNERS = *OFF
Dynamic partners are not permitted. This means that it is only possible to access partner systems which are entered in the partner list and are addressed via the partner name. Transfer requests with partners which are not entered in the partner list or are entered in the partner list without a name are not permitted.

DYNAMIC-PARTNERS = *ON
Dynamic partners are permitted. This means that transfer requests are also permitted with partner systems which are not entered in the partner list or only have their address entered there.

ADM-PORT =
This option allows you to specify the port number used for remote administration.
Default setting following installation: 11000

ADM-PORT = <u>*UNCHANGED</u>
The previous value is unchanged**.**

ADM-PORT = *STD
The port number is set to the default value 11000.

ADM-PORT = <integer 1..65535>
Specifies a valid port number.

ACTIVE-APPLICATIONS=
This option allows you to activate or deactivate the asynchronous inbound server.
Default setting following installation: *OPENFT,*ADM

ACTIVE-APPLICATIONS = <u>*UNCHANGED</u>
The previous value is unchanged**.**

ACTIVE-APPLICATIONS = *ALL
The asynchronous inbound servers for the openFT, ADM and FTP protocols are activated.

ACTIVE-APPLICATIONS = *NONE
The asynchronous inbound servers for the openFT, ADM and FTP protocols are deactivated.

ACTIVE-APPLICATIONS = list-poss(3): *OPENFT / *ADM / *FTP
You can activate the asynchronous inbound servers for specific protocols (openFT, ADM, and/or FTP), by specifying a comma-delimited list of one or more asynchronous inbound servers listed.
The asynchronous inbound servers for the protocol types that are not in the list are then automatically deactivated.

ACTIVE-APPLICATIONS = *OPENFT
Activates the asynchronous inbound server for requests via the openFT protocol.

ACTIVE-APPLICATIONS = *ADM
Activates the asynchronous inbound server for administration requests.

ACTIVE-APPLICATIONS = *FTP
Activates the asynchronous inbound server for requests via the FTP protocol.

ADM-CONNECTION-LIMIT =
This allows you to specify the maximum number of connections for remote administration.
Default setting following installation: 8

ADM-CONNECTION-LIMIT = <u>*UNCHANGED</u>
The previous value is unchanged.

ADM-CONNECTION-LIMIT = <integer 1..99>
You can enter a value between 1 and 99 here.
The default value after installation is 8.

MONITORING =
Activates or deactivates the monitoring functions.
Default setting following installation: *OFF

MONITORING = <u>*UNCHANGED</u>
The monitoring settings remain unchanged.

MONITORING = *ON
Activates monitoring without changing the filter.

MONITORING = *OFF
Deactivates monitoring.

MONITORING = *PARAMETERS(...)
Selects the options that are to be applied to monitoring.

> **SWITCH =**
> Activates or deactivates monitoring for the selected partners.
> Default setting following installation: *OFF
>
> **SWITCH = <u>*UNCHANGED</u>**
> The previous value is unchanged.

SWITCH = *ON
Activates monitoring.

SWITCH = *OFF
Deactivates monitoring.

PARTNER-SELECTION =
Selects the partners that are to be monitored.
Default setting following installation: *ALL

PARTNER-SELECTION = <u>*UNCHANGED</u>
The previous value is unchanged.

PARTNER-SELECTION = *ALL
All the partners are selected for monitoring.

PARTNER-SELECTION = *NONE
No partner is selected for monitoring. In this event, only certain monitoring data values are populated, see the section “Description of the monitoring values” on page 304.

PARTNER-SELECTION = *OPENFT
All partners which are addressed via the openFT protocol are selected for monitoring.

PARTNER-SELECTION = *FTP
All partners which are addressed via the FTP protocol are selected for monitoring.

REQUEST-SELECTION =
Selects the request types for which monitoring data is to be collected.
Default setting following installation: *ALL

REQUEST-SELECTION = <u>*UNCHANGED</u>
The previous value is unchanged.

REQUEST-SELECTION = *ALL
All requests are selected for monitoring.

REQUEST-SELECTION = *ONLY-SYNC
All synchronous requests are selected for monitoring. Synchronous requests are always issued locally.

REQUEST-SELECTION = *ONLY-ASYNC
All asynchronous requests are selected for monitoring. Requests issued remotely are always regarded as asynchronous.

REQUEST-SELECTION = *ONLY-LOCAL
All locally submitted requests are selected for monitoring.

REQUEST-SELECTION = *ONLY-REMOTE
All remotely submitted requests are selected for monitoring.

ADM-TRAPS =
Specifies the settings for the ADM trap server and the ADM traps.
Default setting following installation: *NONE

ADM-TRAPS = *UNCHANGED
The previous settings remain unchanged.

ADM-TRAPS = *NONE
The ADM trap server is removed from the list, the FTAC transfer admission is deleted and all ADM traps are deactivated.

ADM-TRAPS = *PARAMETERS(...)
Changes the name of the destination, i.e. the ADM trap server and the associated FTAC transfer admission and activates or deactivates selected ADM traps.

DESTINATION =
Here you specify the name of the destination or the ADM trap server together with the corresponding FTAC transfer admission.
Default setting following installation: *NONE

DESTINATION = *UNCHANGED
The name of the ADM trap server and the FTAC transfer admission remain unchanged.

DESTINATION = *NONE
The name of the ADM trap server and the FTAC transfer admission are deleted and thus reset to *NONE.

DESTINATION = *PARAMETERS(...)
Destination to which the ADM traps are to be sent.

PARTNER = *UNCHANGED
The name of the ADM trap server remains unchanged.

PARTNER = <text 1..200 with-low>
Name of the partner system from the partner list or the address of the partner system to which the ADM traps are to be sent. If the partner is not entered in the partner list, it must be specified with the prefix ftadm://, see section “Specifying partner addresses” on page 76.

TRANSFER-ADMISSION =
FTAC transfer admission for accessing the ADM trap server.

TRANSFER-ADMISSION = *UNCHANGED
The FTAC transfer admission of the ADM trap server remains unchanged.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <
c-string 8..32 with-low> / <x-string15..64>
The FTAC functionality is used on the remote system. Only the transfer admission defined in the admission profile may be used.

SELECTION =
Activates or deactivates specific ADM traps.
Default setting following installation: *NONE

SELECTION = <u>*UNCHANGED</u>
The previous value is unchanged.

SELECTION = *NONE
Deactivates all ADM traps.

SELECTION = *ALL
Activates all ADM traps.

SELECTION = *PARAMETERS(...)
Activates or deactivates selected ADM traps.

FT-STATE =
Activates or deactivates the sending of traps on FTSTART / FTSTOP and if openFT is terminated abnormally.
Default setting following installation: *OFF

FT-STATE = <u>*UNCHANGED</u>
The previous value is unchanged.

FT-STATE = *OFF
Deactivates the traps for FT-STATE.

FT-STATE = *ON
Activates the traps for FT-STATE.

PARTNER-STATE =
Activates or deactivates the sending of traps when the status of partners changes.
Default setting following installation: *OFF

PARTNER-STATE = <u>*UNCHANGED</u>
The previous value is unchanged.

PARTNER-STATE = *OFF
Deactivates the traps for PARTNER-STATE.

PARTNER-STATE = *ON
Activates the traps for PARTNER-STATE.

PARTNER-UNREACHABLE =
Activates or deactivates the sending of the trap indicating that a partner is unreachable.
Default setting following installation: *OFF

PARTNER-UNREACHABLE = <u>*UNCHANGED</u>
The previous value is unchanged.

PARTNER-UNREACHABLE = *OFF
Deactivates the "partner unreachable" trap.

PARTNER-UNREACHABLE = *ON
Activates the "partner unreachable" trap.

REQUEST-QUEUE-STATE =
Activates the sending of traps referring to the filling level of the request queue, i.e. whether traps are sent if the filling level has exceeded the 85% threshold or fallen below the 80% threshold.
Default setting following installation: *OFF

REQUEST-QUEUE-STATE = <u>*UNCHANGED</u>
The previous value is unchanged.

REQUEST-QUEUE-STATE = *OFF
Deactivates the traps if the filling level falls outside the thresholds.

REQUEST-QUEUE-STATE = *ON
Activates the traps if the filling level falls outside the thresholds.

TRANSFER-SUCCESS =
Activates or deactivates the sending of the trap indicating that an FT request was completed successfully.
Default setting following installation: *OFF

TRANSFER-SUCCESS = <u>*UNCHANGED</u>
The previous value is unchanged.

TRANSFER-SUCCESS = *OFF
Deactivates the trap for TRANSFER-SUCCESS.

TRANSFER-SUCCESS = *ON
Activates the trap for TRANSFER-SUCCESS.

TRANSFER-FAILURE =
Activates or deactivates the sending of the trap indicating that an FT request was aborted.
Default setting following installation: *OFF

TRANSFER-FAILURE = <u>*UNCHANGED</u>
The previous value is unchanged.

TRANSFER-FAILURE = *OFF
Deactivates the trap for TRANSFER-FAILURE.

TRANSFER-FAILURE = *ON
Activates the trap for TRANSFER-FAILURE.

ENCRYPTION-MANDATORY =
Controls the system-wide obligation for user data encryption. This setting applies for transfer and administration requests.
Default setting following installation: *NO

ENCRYPTION-MANDATORY = *UNCHANGED
The setting remains unchanged.

ENCRYPTION-MANDATORY = *NO
Deactivates the system-wide obligation for user data encryption. If encryption is required, this must be specified explicitly in the request.

ENCRYPTION-MANDATORY = *INBOUND
Activates the obligation for inbound encryption:
Inbound requests must transfer the user data in encrypted form, otherwise they are rejected.

ENCRYPTION-MANDATORY = *OUTBOUND
Activates the obligation for outbound encryption, i.e.:
Outbound requests transfer the user data in encrypted form, even if no encryption was called for in the request (e.g. FTACOPY, program interface, etc.).

ENCRYPTION-MANDATORY = (*INBOUND,*OUTBOUND)
Activates the obligation for inbound and outbound encryption, i.e:
Inbound requests must be transferred in encrypted form, otherwise they are rejected.
Outbound requests transfer the user data in encrypted form, even if no encryption was called for in the request.

- System-wide mandatory encryption may be activated only if openFT-CR is installed. Deactivation with ENCRYPTION-MANDATORY=*NO is, on the other hand, permitted even if openFT-CR is no (longer) installed.
- When mandatory inbound encryption is activated, inbound FTAM requests and inbound FTP requests are rejected.
  mandatory Outbound FTP requests are, however, permitted.
- File management requests are executed in unencrypted format irrespective of the specification in ENCRYPTION-MANDATORY.

DELETE-LOGGING =
Controls the settings for deleting log records.

DELETE-LOGGING = *UNCHANGED
The settings for deleting log records remain unchanged.

DELETE-LOGGING = *PARAMETERS(...)
Defines the options for deleting log records.

SWITCH =
Activates or deactivates the automatic deletion of log records.
Default setting following installation: *OFF

SWITCH = *UNCHANGED
The automatic deletion of log records remains activated or deactivated.

SWITCH = *ON
Activates the automatic deletion of log records.

SWITCH = *OFF
Deactivates the automatic deletion of log records.

RETENTION-PERIOD =
Specifies the minimum age of the log records for deletion.
Default setting following installation: 14 days.

RETENTION-PERIOD = *UNCHANGED
The settings remain unchanged.

RETENTION-PERIOD = <integer 0..999 days>
Minimum age of log records for deletion in days. The days are counted back from the deletion time specified in DELETE-TIME. The value 0 deletes all the log records that were written before or at the time of the current day specified in DELETE-TIME.

REPEAT =
Specifies when deletion is to be repeated.
Default setting following installation: *DAILY

REPEAT = *UNCHANGED
The settings remain unchanged.

REPEAT = *DAILY
The log records are deleted every day.

REPEAT = *WEEKLY(..)
The log records are deleted once a week.

ON = *SUNDAY / *MONDAY / *TUESDAY / *WEDNESDAY / *THURSDAY / *FRIDAY / *SATURDAY
Weekday on which the log records are deleted.

REPEAT = *MONTHLY(..)
The log records are deleted once a month.

ON = 1 / <integer 1..31>
Specific day of the month (1-31). If 29, 30 or 31 is specified as the day of the month but the month has fewer days, deletion will take place on the last day of the month.

DELETE-TIME =
Specifies the time at which the log records are to be deleted.
Default setting following installation: 00:00

DELETE-TIME = *UNCHANGED
The setting remains unchanged.

DELETE-TIME = <time 1..8>
Time (local time at which the log records are to be deleted. Due to the nature of the system, the delete function can be performed up to 5 minutes after this time. You enter the time in the format *hh:mm:ss*, e.g. 14:30:10.

RECOVERY-OUTBOUND=
Specifies the global restart function for outbound requests.
Default setting following installation: *ON

RECOVERY-OUTBOUND=*UNCHANGED
The present restart function setting for outbound requests remains unchanged.

RECOVERY-OUTBOUND=*ON
The restart function for outbound requests is activated for all partner systems for which there is no partner-specific setting.

ℹ The restart function for outbound requests is valid only, if the restart function for inbound requests is activated by the foreign system.

RECOVERY-OUTBOUND=*OFF
The restart function for outbound requests is deactivated for all partner systems for which there is no partner-specific setting.

RECOVERY-INBOUND=
This operand specifies the global restart function for inbound requests of partner systems which have activated the recovery for outbound requests.
Default setting following installation: *ON

RECOVERY- INBOUND =*UNCHANGED
The present restart function setting for inbound requests remains unchanged.

RECOVERY- INBOUND =*ON
The restart function for inbound requests is globally activated.

RECOVERY- INBOUND =*OFF
The restart function for inbound requests is globally deactivated.

Example

The maximum number of tasks to be executed in parallel is to be 3 and the maximum number of transport connections to be set up is to be 10:

```
FTMODOPT PROCESS-LIMIT=3,CONNECTION-LIMIT=10
```

4.22 FTMODPRF
Modify 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

The command FTMODPRF can be used by any FTAC user to modify his/her admission profile. In a privileged admission profile, an FTAC user can only modify the operands TRANSFER-ADMISSION and PRIVILEGED.

Under certain circumstances, the FTAC administrator may modify external admission profiles:

– The FTAC administrator possesses the SU privilege. He/She can then modify profiles for other user IDs without restriction.

– If the FTAC administrator does not possess the SU privilege but specifies ACCOUNT and PASSWORD in the USER-ADMISSION parameter, then he/she may also modify admission profiles. The TRANSFER-ADMISSION is only valid for as long as the current password for the user ID corresponds to the one defined in the profile.

– If the FTAC administrator does not possess the SU privilege and also does not specify the user’s password, he/she may not modify the transfer admission of a foreign user profile.

When the FTAC administrator neither possesses SU privilege nor has specified the account number and password, the profile is prohibited after a modification and must be released by the user. Modification of the privilege is excluded from this: in this case the profile is not locked.

As soon as an admission profile is modified, the timestamp of the last modification is also updated. You can see the timestamp with FTSHWPRF INF=*ALL (LAST-MODIF). The timestamp is updated even if you do not change the properties of the profile, i.e. if you enter FTMODPRF with the parameter NAME without specifying other parameters.

Format

(part 1 of 3)

```
FTMODPRF

 NAME = *ALL / *STD / <alphanum-name 1..8>
,PASSWORD = *NONE / <alphanum-name 1..8>
,SELECT-PARAMETER = *OWN / *PARAMETERS(...)

   *PARAMETERS(...)
     |  TRANSFER-ADMISSION = *ALL / *NOT-SPECIFIED / <alphanum-name 8..32> /
     |                               c-string 8..32 with-low> / <x-string 15..64>
     | ,OWNER-IDENTIFICATION = *OWN / *ALL / <name 1..8>

,NEW-NAME = *OLD / *STD / <alphanum-name 1..8>
,TRANSFER-ADMISSION = *UNCHANGED / *NOT-SPECIFIED / *OLD-ADMISSION(...) /
                          <alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) /
                          <x-string 15..64>(...)

   *OLD-ADMISSION(...)
     |  VALID = *UNCHANGED / *YES / *NO
     | ,USAGE = *UNCHANGED / *PRIVATE / *PUBLIC
     | ,EXPIRATION-DATE = *UNCHANGED / *NOT-RESTRICTED / <date 8..10>

   <alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...)
     |  VALID = *YES / *NO / *UNCHANGED
     | ,USAGE = *PRIVATE / *PUBLIC / *UNCHANGED
     | ,EXPIRATION-DATE = *NOT-RESTRICTED / <date 8..10> / *UNCHANGED

,PRIVILEGED = *UNCHANGED / *NO / *YES
,IGNORE-MAX-LEVELS = *UNCHANGED / *NO / *YES / *PARAMETERS(...)

   *PARAMETERS(...)
     |  OUTBOUND-SEND = *UNCHANGED / *NO / *YES
     | ,OUTBOUND-RECEIVE = *UNCHANGED / *NO / *YES
     | ,INBOUND-SEND = *UNCHANGED / *NO / *YES
     | ,INBOUND-RECEIVE = *UNCHANGED / *NO / *YES
     | ,INBOUND-PROCESSING = *UNCHANGED / *NO / *YES
     | ,INBOUND-MANAGEMENT = *UNCHANGED / *NO / *YES
```

(part 2 of 3)

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

   *PARAMETERS(...)
     |    USER-IDENTIFICATION = *OWN / <name 1..8>
     |   ,ACCOUNT = *OWN / *NOT-SPECIFIED / *NONE / <alphanum-name 1..40> / <c-string 1..40>
     |   ,PASSWORD = *OWN / *NOT-SPECIFIED / <alphanum-name 1..8> / *NONE

,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..59> /
                  <c-string 1..512 with-low> /  *EXPANSION(...)

   *EXPANSION(...)
     |    PREFIX = <filename 1..58> / <filename-prefix 2..50> / <c-string 1..511 with-low>

,FILE-PASSWORD = *UNCHANGED / *NOT-RESTRICTED / *NONE / <alphanum-name 1..8>
```

(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..40> / <c-string 1..40>
   |  ,PASSWORD = *SAME / *NOT-RESTRICTED / *NONE / <alphanum-name 1..8>

,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 / *REMOTE-ADMINISTRATION

,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 standard admission profile for your user ID or, as FTAC administrator, the standard admission profile of the selected user ID.

NAME = <alphanum-name 1..8>
Modifies the admission profile with this name.

PASSWORD =
FTAC password which authorizes you to use FTAC commands on your user ID, if such a password has been defined in your admission set.

PASSWORD = *NONE
No FTAC password is required.

PASSWORD = <alphanum-name 1..8>
This FTAC password is required.

SELECT-PARAMETER =
Specifies a transfer admission. You will then modify the admission profile which has this transfer admission.

SELECT-PARAMETER = *OWN
Modifies your own admission profile.

SELECT-PARAMETER = *PARAMETERS(...)
Specifies the selection criteria for the profiles which you wish to modify.

TRANSFER-ADMISSION =
Entering the TRANSFER-ADMISSION here makes it a selection criterion for the admission profiles which you wish to modify.

TRANSFER-ADMISSION = *ALL
All your admission profiles are to be modified, irrespective of the transfer admission.

TRANSFER-ADMISSION = *NOT-SPECIFIED
Only admission profiles without a defined transfer admission are to be modified. In the case of a standard admission profile, the transfer admission is never assigned, because this is addressed using the user ID and the user password.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
The admission profile with this transfer admission is to be modified.

OWNER-IDENTIFICATION =
You can use the owner of an admission profile as a selection criterion for access to a profile to be modified.

OWNER-IDENTIFICATION = *OWN
Modifies your own admission profile.

OWNER-IDENTIFICATION = *ALL
The FTAC administrator can access the profiles of all users. The FTAC user is not permitted to make this entry.

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 standard admission profile for the user ID. If the admission profile previously had a transfer admission, you must also specify TRANSFER-ADMISSION=*NOT-SPECIFIED.

NEW-NAME = <alphanum-name 1..8>
New name of the admission profile. This name must be unique among all the admission profiles on your user ID. If an admission profile with this name already exists, FTAC rejects the command with the following message:

```
FTC0100   COMMAND REJECTED. FT-PROFILE ALREADY EXISTS
```

The command FTSHWPRF (see page 321) can be used to obtain information on the already existing name. For this information, it suffices to enter FTSHWPRF without parameters.

TRANSFER-ADMISSION =
Modifies the transfer admission which is associated with the admission profile selected. You must ensure that the transfer admission is unique within your openFT system. If the transfer admission which you have selected already exists, FTAC rejects the command with the following message:

```
FTC0101   COMMAND REJECTED. TRANSFER-ADMISSION ALREADY EXISTS
```

The FTAC administrator can also allocate a transfer admission here if he/she modifies the admissions profile of any user ID. If he/she has no TSOSSU privilege, the FTAC administrator must also specify the complete USER-ADMISSION for the affected user ID (USER-IDENTIFICATION, ACCOUNT and PASSWORD).
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 locks the profile, provided that it is not a profile that you are converting to a standard 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 standard admission profile.

VALID = <u>*UNCHANGED</u>
The value remains unchanged.

VALID = *YES
The transfer admission is valid.

VALID = *NO
The transfer admission is not valid. The profile can be locked with this entry.

USAGE = <u>*UNCHANGED</u>
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 = <u>*UNCHANGED</u>
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. 2017-12-31 or 17-12-31 for 31 December, 2017.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 = <u>*YES</u>
The transfer admission is valid.

VALID = *NO
The transfer admission is not valid. The profile can be locked 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. 2017-12-31 or 17-12-31 for 31 December, 2017..The use of the transfer admission is only possible until the given date.

EXPIRATION-DATE = *UNCHANGED
The value remains unchanged.

PRIVILEGED =
The FTAC administrator can privilege the admission profile of any FTAC user. FT requests which are processed with a privileged admission profile 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 272).

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”.

INBOUND-RECEIVE = *UNCHANGED
The maximum security level which can be reached with the basic function “inbound receive” remains unchanged.

INBOUND-RECEIVE = *NO
The maximum security level which can be reached with the basic function “inbound receive” is determined by the admission set.

INBOUND-RECEIVE = *YES
Disregards your settings for “inbound receive” in the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS. The same applies to the following partial components of the basic function “inbound file management”:

- delete files, as long as the file attributes are set accordingly,
- modify file attributes, if the basic function “inbound file management” was admitted in the admission set or in the admission profile.

INBOUND-PROCESSING = *UNCHANGED
The maximum security level which can be reached with the basic function “inbound processing” remains unchanged.

INBOUND-PROCESSING = *NO
The maximum security level which can be reached with the basic function “inbound processing” is determined by the admission set.

INBOUND-PROCESSING = *YES
For the basic function “inbound processing”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS.

INBOUND-MANAGEMENT = *UNCHANGED
The maximum security level which can be reached with the basic function “inbound file management” remains unchanged.

INBOUND-MANAGEMENT = *NO
The maximum security level which can be reached with the basic function “inbound file management” is determined by the admission set.

INBOUND-MANAGEMENT = *YES
For the basic function “inbound file management”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS. The partial component “modify file attributes” of the basic function “inbound file management” only functions if the basic function “inbound receive” was admitted in the admission set or admission profile.

USER-ADMISSION =
User ID under which the modified admission profile is saved. FT requests which use this profile access the entered user ID in the local system.
As an FTAC user you can only specify your own user ID here.
If the FTAC administrator has created an admission profile for a user without specifying the access data (see the FTCREPRF command, page 119), the user must, if necessary, enter the account and password in the operands ACCOUNT and PASSWORD described below before the profile can be used.

USER-ADMISSION = *UNCHANGED
The USER-ADMISSION of this admission profile remains unchanged.

USER-ADMISSION = *OWN
For USER-IDENTIFICATION and ACCOUNT, the specifications are taken from the current LOGON authorization. A z/OS password is only taken from your LOGON authorization when an FT request accesses the admission profile.

Admission profiles in which USERID, ACCOUNT and/or PASSWORD in USER-ADMISSION are set to their default values via *OWN cannot be used for preprocessing, postprocessing or follow-up processing. For preprocessing and postprocessing, these parameters must be explicitly assigned a value in USER-ADMISSION. For follow-up processing, a specification in PROCESSING-ADMISSION is also possible.

USER-ADMISSION = *PARAMETERS(...)
Specifies the individual components of the user ID.
This allows you, for example, to ensure that FT requests which use this admission profile are kept under a different account number from the currently valid account number. Another application is to specify a password in the admission profile. FT requests which use this admission profile will then only function if the current LOGON password corresponds to this preset password.

USER-IDENTIFICATION =
Your user ID in z/OS

USER-IDENTIFICATION = <u>*OWN</u>
The user ID is taken from your LOGON authorization.

USER-IDENTIFICATION = <name 1..8>
User ID with which the profile is to be associated. As FTAC administrator you may also specify foreign user IDs.

ACCOUNT =
Account number under which an FT request is to be kept when it uses this admission profile.

ACCOUNT = <u>*OWN</u>
The account number is taken from the current LOGON authorization.

ACCOUNT = *NOT-SPECIFIED
No account number is defined.
The account number is to be specified by the owner of the admission profile. This function permits the FTAC administrator to set up profiles for user IDs whose account numbers he/she does not know.

ACCOUNT = *NONE
Has the same effect as ACCOUNT = *NOT-SPECIFIED.

ACCOUNT = <alphanum-name 1..40> / <c-string 1..40>
An FT request should be kept under the account number specified when it accesses this admission profile. You can enter any account number which is associated with your user ID.
You can also specify accounting information containing the account number which is to be used.

PASSWORD =
Password which an FT request is to use when it works with this admission profile.

PASSWORD = <u>*OWN</u>
When an FT request refers to this admission profile, FTAC uses the password valid at that moment. This prevents you from having to modify the admission profile if the BS2000 password is changed.

Admission profiles in which PASSWORD is set to its default value via *OWN cannot be used for preprocessing, postprocessing or follow-up processing. For preprocessing and postprocessing, this parameter must be explicitly assigned a value. For follow-up processing, a specification in PROCESSING-ADMISSION is also possible.

PASSWORD = *NOT-SPECIFIED
The password is specified by the owner of the admission profile. This function permits the FTAC administrator to set up profiles for foreign user IDs whose access data he/she does not know.

PASSWORD = <alphanum-name 1..8>
When an FT request accesses the admission profile, the specified password is compared with the current LOGON password. If the two do not correspond, the FT request is rejected.

PASSWORD = *NONE
No password is required for the user ID.

INITIATOR =
Determines if initiators from local and/or remote systems are permitted to use this admission profile for their FT requests.

INITIATOR = *UNCHANGED
The settings in this admission profile remain unchanged,

INITIATOR = *REMOTE
This admission profile may only be used for FT requests by initiators from remote systems.

INITIATOR = *LOCAL
This admission profile may only be used for FT requests by initiators from the local system.

INITIATOR = (*LOCAL,*REMOTE)
This admission profile may be used by initiators from local and remote systems.

TRANSFER-DIRECTION =
Determines which transfer direction may be used with this admission profile.

ℹ 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 = <u>*UNCHANGED</u>
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 “Specifying partner addresses” on page 76. 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 = <u>*UNCHANGED</u>
The specification for MAX-PARTNER-LEVEL in this admission set remains unchanged.

MAX-PARTNER-LEVEL = *NOT-RESTRICTED
If FT requests are processed with this admission profile, then the highest accessible security level is determined by the admission set.

MAX-PARTNER-LEVEL = <integer 0..100>
All partner systems which have this security level or lower can be communicated with.

When you set MAX-PARTNER-LEVEL=0, you prevent access to the admission profile (for the time being). No FT request can then be processed with this admission profile.

FILE-NAME =
Determines which files or library members under your user ID may be accessed by FT requests that use this admission profile.

FILE-NAME = *UNCHANGED
The specifications for FILE-NAME in this admission profile remain unchanged.

FILE-NAME = *NOT-RESTRICTED
The admission profile permits unrestricted access to all files and library members of the user ID.

FILE-NAME = <filename 1..59> / <c-string 1..512 with-low>
Only the specified file may be accessed. However, openFT is also able to generate unique filenames automatically, thus providing an easy way of avoiding conflicts. This is done by specifying the string %UNIQUE at the end of the filename which is predefined here. When follow-up processing is specified, this file can be referenced with %FILENAME, %FILN or %FILX.
You can also directly specify file transfer with pre- and postprocessing here by entering the pipe symbol ’|’ followed by a command.

FILE-NAME =*EXPANSION(PREFIX = <filename 1..58> / <filename-prefix 2..50> / <c-string 1..511 with-low>)
Restricts access to a number of files which all begin with the same prefix. If a *filename* is entered in an FT request which uses this admission profile, FTAC sets the *prefix* defined with EXPANSION in front of this filename. The FT request is then permitted to access the file *PrefixFilename*.

Example

– If PREFIX=STEVEN. is specified, a FT request in which the FILE-NAME=MILLER is specified accesses the file STEVEN.MILLER.

– If PREFIX=TOOLS.CLIST/ is specified, a FT request in which FILE-NAME=MEMBER01 was specified, then accesses the file TOOLS.CLIST(MEMBER01).

Please note that the part of a filename which is specified in the file transfer command still has to be of the type <filename>.

If you want to perform file transfer with pre- or postprocessing, 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 postprocessing 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 81.

The maximum length of the entire pre- or postprocessing command is limited to the maximum length of the file name. If several commands are specified, then they must be separated by a semicolon (‘;’).

Example

```
FILE-NAME = *EXP(C‘|Command1;Command2;Command3; ...‘)
```

If you specify a name prefix that starts with a pipe character with *EXP(PREFIX=...), the preprocessing or postprocessing command of the FT request must not contain any semicolons. If the preprocessing or postprocessing command nevertheless contains semicolons, it must be enclosed in '...' (single quotes) .

Special cases

– In the case of admission profiles which are to be used exclusively for the ftexec command you must specify a filename or filename prefix that starts with the character string ‘|ftexecsv’ (see FTCREPRF, page 119).

– 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 page 119).

FILE-PASSWORD =
You can enter a password for files into the admission profile. The FTAC functionality then only permits access to files which are protected with this password and to unprotected files. When a FILE-PASSWORD is specified in an admission profile, the password may no longer be specified in an FT request which uses this admission profile. This allows you to permit access to certain files to users in remote systems, without having to disclose the file passwords.

FILE-PASSWORD = *UNCHANGED
The specifications for FILE-PASSWORD in this admission profile remain unchanged.

FILE-PASSWORD = *NOT-RESTRICTED
Permits access to all files. If a password is set for a file, then it must be specified in the transfer request.

FILE-PASSWORD = *NONE
Only permits access to files without file passwords.

FILE-PASSWORD = <alphanum-name 1..8>
Only permits access to files which are protected with the password specified and to unprotected files. The password which has already been specified in the profile may not be repeated in the transfer request. PASSWORD=*NONE would be entered in this case!

PROCESSING-ADMISSION =
You can enter a user ID in your z/OS system. Any follow-up processing of an FT request will be executed under this user ID. With PROCESSING-ADMISSION in the admission profile, you do not need to disclose your LOGON authorization to partner systems for follow-up processing.

i Admission profiles in which ACCOUNT and/or PASSWORD in USER-ADMISSION are set to their default values via *OWN cannot be used for follow-up processing. For follow-up processing, these parameters must be explicitly assigned a value either in USER-ADMISSION or in PROCESSING-ADMISSION.

PROCESSING-ADMISSION = *UNCHANGED
The PROCESSING-ADMISSION in this admission profile remains unchanged.

PROCESSING-ADMISSION = *SAME
For the PROCESSING-ADMISSION, the values of the USER-ADMISSION are used. If *SAME is entered here, then any FT request which uses this profile must also contain PROCESSING-ADMISSION=*SAME or PROCESSING-ADMISSION= *NOT-SPECIFIED.

PROCESSING-ADMISSION = *NOT-RESTRICTED
FT requests which use this admission profile may contain any PROCESSING-ADMISSION.

PROCESSING-ADMISSION = *PARAMETERS(...)
You can also enter the individual components of the user ID. This allows follow-up processing using this admission profile and started from FT requests to be charged under a different account number, for example. Or, a password can be set in the admission profile. Follow-up processing for FT requests which use this admission profile will then only function if their current LOGON password corresponds to the pre-set password.

USER-IDENTIFICATION =
User ID under which the follow-up processing is to be executed.

USER-IDENTIFICATION = *SAME
The USER-IDENTIFICATION is taken from the USER-ADMISSION.

USER-IDENTIFICATION = *NOT-RESTRICTED
The admission profile does not restrict the user ID under which the follow-up processing is to be executed.

USER-IDENTIFICATION = <name 1..8>
FT requests which are processed with this admission profile are only permitted follow-up processing under this user ID. If another user ID is entered here, the parameter PASSWORD must also be entered. PASSWORD=*SAME is then not valid.

ACCOUNT =
Specifies the account number for the follow-up processing.

ACCOUNT = *SAME
The account number is taken from the USER-ADMISSION.

ACCOUNT = *NOT-RESTRICTED
The account number may be specified in FT requests that work with the admission profile. The admission profile does not restrict the account for follow-up processing.

ACCOUNT = *NONE
The account number is used which is defined as the default account number of the user ID specified at the time the admission profile is used.

ACCOUNT = <alphanum-name 1..40> / <c-string 1..40>
Follow-up processing is to be settled under this account number.
You can also specify account information containing the account number to be used.

PASSWORD =
Specifies, where applicable, the z/OS password for the user ID under which the follow-up processing is to be executed. Here, you can enter a PASSWORD when the user ID in question doesn’t have such a password (yet).

PASSWORD = <u>*SAME</u>
The value *SAME is only valid if the PROCESSING-ADMISSION refers to your own user ID. If PASSWORD=*OWN is entered on USER-ADMISSION, then the BS2000 password valid at the time of the request is used for the PROCESSING-ADMISSION.

PASSWORD = *NOT-RESTRICTED
The password may be specified for FT requests which work with the admission profile. The admission profile does not restrict the password for follow-up processing.

PASSWORD = *NONE
FT requests which use this admission profile can only initiate follow-up processing on user IDs without a password.

PASSWORD = <alphanum-name 1..8>
FT requests which use the admission profile may only initiate follow-up processing on user IDs which are protected with this password.

SUCCESS-PROCESSING =
Restricts the follow-up processing which an FT request is permitted to initiate in your system after a successful data transfer.

SUCCESS-PROCESSING = <u>*UNCHANGED</u>
The specifications for SUCCESS-PROCESSING in this admission profile remain unchanged.

SUCCESS-PROCESSING = *NOT-RESTRICTED
In FT requests which use this admission profile the operand SUCCESS-PROCESSING may be used without restriction.

SUCCESS-PROCESSING = *NONE
The admission profile does not permit follow-up processing after successful data transfer.

SUCCESS-PROCESSING = <c-string 1..1000 with-low>
BS2000 commands which are executed in the local system after successful data transfer. The individual commands must be separated by a semicolon (;). If a character string is enclosed by single or double quotes (’ or ”) within a command sequence, openFT does not interpret any semicolons within this character string as a separator.

SUCCESS-PROCESSING = *EXPANSION(...)
If a SUCCESS-PROCESSING was specified in an FT request which uses this admission profile, FTAC adds the prefix or suffix specified here to this command. As follow-up processing, the command which has been thus expanded is then executed.

If a suffix or prefix is defined at this point, then no command sequence for the follow-up processing may be specified in FT requests which use this admission profile. This makes the setting of prefixes and suffixes mandatory.

PREFIX = <u>*UNCHANGED</u>
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 = <u>*UNCHANGED</u>
The specifications for the follow-up processing suffix in this admission profile remain unchanged.

SUFFIX = *NOT-RESTRICTED
Follow-up processing is not restricted by a suffix.

SUFFIX = <c-string 1..999 with-low>
The specified prefix is set after a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the suffix is executed as follow-up processing.

Example

If PREFIX='SEND ' and SUFFIX=',USER(USER1)' is specified and SUCC='''FILE TRANSFER OK''' is defined in the FT request, FT executes the command "SEND 'FILE TRANSFER OK',USER(USER1)" for follow-up processing.

FAILURE-PROCESSING =
Restricts the follow-up processing which an FT request is permitted to initiate in your system after a failed data transfer.

FAILURE-PROCESSING = *UNCHANGED
The specifications for FAILURE-PROCESSING in this admission profile remain unchanged.

FAILURE-PROCESSING = *NOT-RESTRICTED
In FT requests which use this admission profile the operand FAILURE-PROCESSING may be used without restriction.

FAILURE-PROCESSING = *NONE
The admission profile does not permit follow-up processing after failed data transfer.

FAILURE-PROCESSING = <c-string 1..1000 with-low>
z/OS commands which are executed in the local system after failed data transfer. Individual commands must be preceded by a slash (/). The individual commands must be separated by a semicolon (;). If a character string is enclosed by single or double quotes (’ or ”) within a command sequence, openFT does not interpret any semicolons within this character string as a separator.

FAILURE-PROCESSING = *EXPANSION(...)
If a FAILURE-PROCESSING was specified in an FT request which uses this admission profile, FTAC adds the prefix or suffix specified here to this command. As follow-up processing, the command which has been thus expanded is then executed.

If a suffix or prefix is defined at this point, then no command sequence for the follow-up processing may be specified in FT requests which use this admission profile. This makes the setting of prefixes and suffixes mandatory.

PREFIX = *UNCHANGED
The specifications for the follow-up processing prefix in this admission profile remain unchanged.

PREFIX = *NOT-RESTRICTED
Follow-up processing is not restricted by a prefix.

PREFIX = <c-string 1..999 with-low>
The specified prefix is set in front of a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the prefix is executed as follow-up processing.

SUFFIX = *UNCHANGED
The specifications for the follow-up processing suffix in this admission profile remain unchanged.

SUFFIX = *NOT-RESTRICTED
Follow-up processing is not restricted by a suffix.

SUFFIX = <c-string 1..999 with-low>
The specified prefix is set after a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the suffix is executed as follow-up processing.

WRITE-MODE =
Determines the WRITE-MODE which is valid for this FT request. WRITE MODE is only effective if the receive file is in the same system as the admission profile definition.

WRITE-MODE = *UNCHANGED
The specifications for WRITE-MODE in this admission profile remain unchanged.

WRITE-MODE = *NOT-RESTRICTED
In an FT request which accesses this admission profile, WRITE-MODE may be used without restrictions.

WRITE-MODE = *NEW-FILE
In the FT request, *NEW-FILE, *REPLACE-FILE or *EXTEND-FILE may be entered for WRITE-MODE. If the receive file already exists, the transfer will be rejected.

WRITE-MODE = *REPLACE-FILE
In the FT request of openFT partners, only *REPLACE-FILE or *EXTEND-FILE may be entered for WRITE-MODE. With ftp partners, *NEW-FILE may also be entered if the file does not yet exist.

WRITE-MODE = *EXTEND-FILE
In the FT request, only *EXTEND-FILE may be entered for WRITE-MODE.

FT-FUNCTION =
This operand permits the restriction of the profile validity to certain FT functions (=file transfer and file management functions).

FT-FUNCTION = *UNCHANGED
The previous scope of the FT functions remains unchanged.

FT-FUNCTION = *NOT-RESTRICTED
The full scope of FT functions is available with the exception of the “remote administration” function (*REMOTE-ADMINISTRATION). This must be activated explicitly.

FT-FUNCTION = (*TRANSFER-FILE, *MODIFY-FILE-ATTRIBUTES, *READ-DIRECTORY, *FILE-PROCESSING, *REMOTE-ADMINISTRATION)
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 “preprocessing” and “postprocessing” 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 postprocessing (pipe symbol “**|**”) are to be possible with this FTAC profile.

***REMOTE-ADMINISTRATION**
The admission profile is allowed to be used for the "remote administration" function. This allows a remote administrator to administer the openFT instance using this profile. *REMOTE-ADMINISTRATION may only be specified by the FT administrator or FTAC administrator.

USER-INFORMATION =
Specifies a text in the admission profile. This text can be displayed with the FTSHWPRF command.

USER-INFORMATION = *UNCHANGED
Any existing text remains unchanged.

USER-INFORMATION = *NONE
Any existing text is deleted.

USER-INFORMATION = <c-string 1..100 with-low>
The character string entered is accepted as user information.

DATA-ENCRYPTION =
Specifies whether user data with this profile must be transferred in encrypted form.

DATA-ENCRYPTION = *UNCHANGED
The encryption option should remain unchanged.

DATA-ENCRYPTION = *NOT-RESTRICTED
The encryption option for user data is not restricted. File transfer requests with encryption and file transfer requests without encryption are both accepted

DATA-ENCRYPTION = *NO
Only file transfer requests that do not have encrypted user data are accepted, i.e. requests with encryption are rejected. If the request is made in a BS2000 or z/OS, DATA-ENCRYPTION=*NO must be specified there in the NCOPY request.

DATA-ENCRYPTION = *YES
Only file transfer requests that have encrypted user data are accepted, i.e. requests without encryption are rejected. If the request is made in a BS2000 or z/OS, for example, then DATA-ENCRYPTION=*YES must be specified there in the NCOPY request.

When using restrictions for FILE-NAME, SUCCESS-PROCESSING and FAILURE-PROCESSING, keep in mind that

- a restriction for follow-up processing must always be made for SUCCESS- and FAILURE-PROCESSING. Otherwise, it is possible that users will avoid this step.
- PREFIX of FILE-NAME, SUCCESS-PROCESSING and FAILURE-PROCESSING must correspond,
  e.g. FILE-NAME = *EXP(XYZ.),SUCC = *EXP(’PR DSNAME(XYZ.’,')')

Example

After Steven Miller has created an admission profile with the name *profile1*, which permits other users access to his user ID without the LOGON authorization, he decides he wants to restrict this profile so that only FT accesses are possible to files which begin with the prefix *BRANCH*.

The required command is:

```
FTMODPRF␣NAME=PROFIL1,
           FILE-NAME=*EXPANSION(PREFIX=BRANCH.)
```

A possible short form of this command is:

```
FTMODPRF␣PROFIL1,FILE-N=(PRE=BRANCH.)
```

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

4.23 FTMODPTN Modify partner properties in the partner list

Note on usage

User group: FT administrator

Functional description

This command can be used to modify the characteristics of a partner that is already entered in the partner list. When changing the partner address, please note that an openFT partner cannot be changed to an FTP partner and vice versa.

You can remove an entered dynamic partner from the partner list by setting all the properties to the default values for free dynamic partners by means of the FTMODPTN command. The default values are the same as the default values in the FTADDPTN command with the exception of the SECURITY-LEVEL operand which must be set to *BY-PARTNER-ATTRIBUTES.

Similarly, you can add a free dynamic partner to the partner list by setting at least one of its attributes to a value other than the default. This is possible if PARTNER does not reference a partner list entry and PARTNER-ADDRESS is not specified.

If a partner name for which there is not yet a partner list entry is specified for PARTNER and PARTNER-ADDRESS is also specified, a new named partner list entry is created. This function is intended for the re-import of exported partner entries. To explicitly create new partner entries, you should use FTADDPTN.

Format

```
FTMODPTN
----------------------------------------------------------------------------
 PARTNER = *ALL / <text 1..200 with-low>
,STATE = *UNCHANGED / *PARAMETERS(...)
    *PARAMETERS(...)
     |  OUTBOUND = *UNCHANGED / *ACTIVE(...) / *DEACT
     |          |  *ACTIVE(...)
     |          |     AUTOMATIC-DEACT = *NO / *YES
     |  ,INBOUND = *UNCHANGED / *ACTIVE / *DEACT
,SECURITY-LEVEL = *UNCHANGED / *STD / *BY-PARTNER-ATTRIBUTES / <integer 1..100>
,PARTNER-ADDRESS = *UNCHANGED / <text 1..200 with-low>
,TRACE = *UNCHANGED / *BY-FT-OPTIONS / *ON / *OFF
,IDENTIFICATION = *UNCHANGED / *STD / <composed-name 1..64> / <c-string 1..64 with-low>
,SESSION-ROUTING-INFO = *UNCHANGED / *NONE / *IDENTIFICATION / <alphanum-name 1..8>
,PARTNER-CHECK = *UNCHANGED / *BY-FT-OPTIONS / *STD / *TRANSPORT-ADDRESS
,AUTH-MANDATORY = *UNCHANGED / *NO / *YES
,PRIORITY= *UNCHANGED / *NORMAL / *LOW / *HIGH
,REQUEST-PROCESSING = *UNCHANGED / *STD / *SERIAL
,RECOVERY-OUTBOUND = *UNCHANGED / *ON / *OFF/ *BY-FT-OPTIONS
```

Operands

PARTNER =

Specifies the partner system.

PARTNER = *ALL

The specified changes are to be implemented for all partner systems defined in the partner list. This specification is only meaningful in conjunction with the operands STATE, SECURITY-LEVEL, TRACE, PARTNER-CHECK, AUTH-MANDATORY, PRIORITY and REQUEST-PROCESSING.

Particular care is necessary when using PARTNER=*ALL in combination with the SECURITY-LEVEL operand.

The description below refers to a single partner system. If you have selected *ALL, the description applies by analogy for all partner system in the partner list which comply with the relevant selection criteria.

PARTNER = <text 1..200 with-low>
Specifies either the name of the partner system from the partner list or the address of the partner system (see section “Specifying partner addresses” on page 76).

STATE =
Controls the state of the partner system (activated, deactivated).

STATE = *UNCHANGED
The state is unchanged.

STATE = *PARAMETERS(...)
Specifies the settings for locally submitted file transfer requests (outbound) and for remotely submitted file transfer requests.

OUTBOUND =
Specifies the setting for locally submitted file transfer requests to the partner system.

OUTBOUND = *UNCHANGED
The state of locally submitted FT requests is unchanged.

OUTBOUND = *ACTIVE(...)
Locally submitted file transfer requests to the partner system are processed.

AUTOMATIC-DEACT =
Defines if repeated attempts to establish a connection with this partner system should result in a deactivation of the partner system after multiple attempts.

AUTOMATIC-DEACT = *NO
Unsuccessful attempts to establish a connection with this partner system do not lead to its deactivation.

AUTOMATIC-DEACT = *YES
Repeated unsuccessful attempts to establish a connection with this partner system lead to its deactivation. If locally submitted file transfer requests to the partner system are to be executed again after this, the system must be activated explicitly (with OUTBOUND=*ACTIVE).

OUTBOUND = *DEACT
Locally submitted file transfer requests to the partner system are initially not processed (not started) but are stored in the request queue. They are executed only after the partner system has been activated with OUTBOUND=*ACTIVE.

INBOUND =
Specifies the setting for remotely submitted file transfer requests, i.e. requests which were submitted by this partner system.

INBOUND = *UNCHANGED
The state of locally submitted FT requests is unchanged.

INBOUND = *ACTIVE
Remotely submitted file transfer requests from this partner system are processed.

INBOUND = *DEACT
Remotely submitted synchronous file transfer requests from this partner system are rejected. Remotely submitted asynchronous file transfer requests from this partner system are stored there and cannot be processed until the partner system is activated again with INBOUND=*ACTIVE.

SECURITY-LEVEL =
Assigns a security level to a remote system.

SECURITY-LEVEL = *UNCHANGED
The value is unchanged.

SECURITY-LEVEL = *STD
If you set this operand to *STD, a standard security level is assigned to the remote system. This standard security level is defined using the FTMODOPT command. Here you can define a fixed value or make the value attribute-dependent.

SECURITY-LEVEL = *BY-PARTNER-ATTRIBUTES
If you set the operand to *BY-PARTNER-ATTRIBUTES then the security level is defined automatically:

– Partners that are authenticated by openFT are assigned the security level 10.
– Partners, known to the transport system (e.g. VTAM or DNS), are assigned the security level 90.
– All other partners are assigned security level 100.

SECURITY-LEVEL = <integer 1..100>
Must be specified if you want to assign a particular security level to the individual partner system.

PARTNER-ADDRESS =
Address of the partner system.

PARTNER-ADDRESS = *UNCHANGED
The address remains unchanged.

PARTNER-ADDRESS = <text 1..200 with-low>
New address for the partner system. For details on the address format, see section “Specifying partner addresses” on page 76.

TRACE =
Trace setting for the partner systems. Trace entries are generated only if the FT trace function is activated by means of an operating parameter (FTMODOPT TRACE=*ON).

TRACE = *UNCHANGED
The current trace setting is unchanged.

TRACE = *BY-FT-OPTIONS
The trace settings specified in the MODIFY-FT-OPTIONS command are used.

TRACE = *ON
Activates the trace for this partner system even if tracing is deactivated for this partner type in the global settings (FTMODOPT). The request-specific trace settings made in FTMODOPT, on the other hand, are taken into account.

TRACE = *OFF
For connections to this partner system, only those trace entries which it is technically impossible to suppress are generated. Trace entries which it is technically impossible to suppress are those which are generated before openFT (BS2000) identifies the partner system

IDENTIFICATION =
The network-wide, unique ID of the openFT instance in the partner system.

IDENTIFICATION = <u>*UNCHANGED</u>
The ID remains unchanged.

IDENTIFICATION = *STD
For openFT and FTADM partners, the partner address or the host name from the partner address is used as the identification. No identification is set for FTPartners.

IDENTIFICATION = <composed-name 1..64> / <c-string 1..64 with-low>
The network-wide, unique instance ID of the openFT instance in the partner system. This ID is used for authenticating partner systems as of openFT V8.1. It is set by the FT administrator of the partner system (in BS2000, by using MODIFY-FT-OPTIONS IDENTIFICATION=, in Unix systems or Windows, by using *ftmodo -id*). The uniqueness of this ID must be based on something other than case-sensitivity. An instance ID may be comprised of alphanumeric characters or special characters. It is advisable to use only the special characters “.”, “-”, “:” or “%”.

The initial character must be alphanumeric or the special character “%”. The “%” character may only be used as an initial character. An alphanumeric character must follow the “.” character. For more details on assigning instance identifications, see section “Instance identifications” on page 88.

The instance identification must not be specified with FTP partners!

SESSION-ROUTING-INFO =
If the partner system is addressed via IDENTIFICATION, but is only accessible via a go-between instance (e.g. an openFTIF gateway), specify here the address information, which the go-between instance will use for re-routing.

SESSION-ROUTING-INFO = <u>*UNCHANGED</u>
The setting remains unchanged.

SESSION-ROUTING-INFO = *NONE
No routing information is used. The session selector can be specified as part of the partner address.

SESSION-ROUTING-INFO = *IDENTIFICATION
Connections to the partner are re-routed via a gateway that uses the instance identification as the address information.

SESSION-ROUTING-INFO = <alphanum-name 1..8>
Connections to the partner are re-routed via a gateway, that uses the specified string as addressing information.

PARTNER-CHECK =
Enables the global settings for sender checking to be modified on a partner-specific basis. These settings are only effective for named openFT partners that do not work with authentication (see openFT manual "Concepts and Functions").

This setting has no meaning for FTP partners and dynamic partner entries.

PARTNER-CHECK = <u>*UNCHANGED</u>
The set value remains unchanged.

PARTNER-CHECK = *BY-FT-OPTIONS
The global settings are valid for the partner.

PARTNER-CHECK = *STD
Disable the expanded sender checking. The transport address of the partner is not checked, even if the expanded sender checking is globally enabled (see the FTMODOPT command).

PARTNER-CHECK = *TRANSPORT-ADDRESS
Enables expanded sender checking. The transport address is checked, even if the expanded sender checking is globally disabled (see the FTMODOPT command).
If the transport address under which the partner is reporting does not correspond to the entry in the partner list, the request is rejected.

AUTH-MANDATORY =
Forces the authentication of a named partner system.

AUTH-MANDATORY = <u>*UNCHANGED</u>
The set value is unchanged.

AUTH-MANDATORY = *NO
Authentication is not forced, i.e. this partner system is not restricted with regard to authentication.

AUTH-MANDATORY = *YES
Authentication is forced, i.e. connections to and from this named partner are only permitted when authentication is provided.

PRIORITY=
This operand allows you to specify the priority of the partner system in respect of processing requests that have the same request priority. This means that the partner priority only applies in the case of requests that have the same request priority, but that are issued to partners with a different partner priority.

PRIORITY = <u>*UNCHANGED</u>
The priority of the partner system with regard to the processing of requests with the same request priority remains unchanged.

PRIORITY = <u>*NORMAL</u>
The partner has normal priority.

PRIORITY = *LOW
The partner has low priority.

PRIORITY = *HIGH
The partner has high priority.

REQUEST-PROCESSING =
You use this option to control whether asynchronous outbound requests to this partner system are always run serially or whether parallel connections are permitted.

REQUEST-PROCESSING = <u>*UNCHANGED</u>
The operating mode to this partner system remains unchanged.

REQUEST-PROCESSING = *STD
Parallel connections to this partner system are permitted.

REQUEST-PROCESSING = *SERIAL
Parallel connections to this partner system are not permitted. If multiple file transfer requests to this partner system are pending, then they are processed serially. A follow-up request is consequently not started until the preceding request has terminated.

RECOVERY-OUTBOUND=
This operand controls the partner-specific restart function (recovery) for asynchronous outbound requests.

RECOVERY-OUTBOUND=<u>*UNCHANGED</u>
The current setting for the recovery of outbound requests remains unchanged.

RECOVERY-OUTBOUND=* ON
The restart function for outbound requests is activated for this partner system. This setting is valid only, if the global restart function of the openFT is activated, too.

RECOVERY-OUTBOUND=* OFF
The restart function for outbound requests is deactivated for this partner system.

RECOVERY-OUTBOUND=* BY-FT-OPTIONS
The global setting for the restart function for outbound requests is valid.

Example 1

The SECURITY-LEVEL for the partner system TEST is set to 99:

```
FTMODPTN PARTNER=TEST,SECURITY-LEVEL=99
```

Example 2

The port number for partner WINDOWS (host name = winhost2) is set to 1100:

```
FTMODPTN WINDOWS,PARTNER-ADDRESS=winhost2:1100
```

4.24 FTMODREQ
Modify request queue

Note on usage

User group: FT user and FT administrator

Functional description

You use the FTMODREQ command to modify the position and priority of your outbound requests within the openFT request queue. You have the option of processing the outbound requests in any order you wish. Newly input requests or requests whose priority changes are put at the end of the request queue for the corresponding priority. If already active requests are repositioned behind waiting outbound requests, the active requests are interrupted if possible in favor of those waiting.

FTMODREQ is only valid for outbound requests.

The sequence of requests with a starting time in the future cannot be modified.

As FT administrator you can modify all requests.

As FT user you can only modify your own requests.

Format

```
FTMODREQ

 TRANSFER-ID = *ALL / <integer 1..2147483647>
,SELECT = *OWN / *PARAMETERS(...)
    *PARAMETERS(...)
     |    OWNER-IDENTIFICATION = *OWN / *ALL / <name 1..8>
     |   ,PARTNER = *ALL / <text 1..200 with-low>
     |   ,FILE = *ALL / <filename 1..59> / <c-string 1..512 with-low>
,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 = *ALL
Modifies outbound requests for all user IDs.
Only the FTAC administrator may use this entry.

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 "Specifying partner addresses" on page 76.

FILE =
Modifies outbound requests which access this file or library member in the local system as a send or receive file. The file or library member name must be entered exactly as in the file transfer request and as it is output using the NSTATUS command. File names with wildcards are not permitted.

FILE = *ALL
The filename is not selected as a criterion for the outbound requests to be modified.

FILE = <filename 1..59> / <c-string 1..512 with-low>
Modifies outbound requests which access this file (DVS/POSIX) in the local system.

QUEUE-POSITION =
New position of the outbound request that is to be modified in the openFT request queue.

QUEUE-POSITION = *UNCHANGED
The position of the outbound request in this user’s openFT request queue remains unchanged.

QUEUE-POSITION = *FIRST
The outbound request is placed in front of all the other requests of the same priority issued by the user in the openFT request queue.

QUEUE-POSITION = *LAST
The outbound request is placed behind all the other requests of the same priority issued by the user in the openFT request queue.

PRIORITY =
Modifies the priority of the FT request.

PRIORITY = *UNCHANGED
The priority of the FT request remains unchanged.

PRIORITY = *NORMAL
The priority of the FT request is set to the normal value

PRIORITY = *HIGH
The FT request is given a high priority.

PRIORITY = *LOW
The FT request is given a low priority.

Example

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

FTMODREQ SELECT=(FILE=FILE2),QUEUE-POS=*FIRST

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

4.25 FTREMPTN Remove remote system from partner list

Note on usage

User group: FT administrator

Functional description

The FTREMPTN command is used to remove a remote system from the partner list of the current openFT instance.

If a partner system is deleted from the partner list then all requests involving this partner system are aborted. FTREMPTN therefore represents a simple way to delete all the requests relating to a given partner. A request to a partner removed with FTREMPTN is eliminated even if the request is already known in the partner system (in the same way as with NCANCEL .. FORCE-CANCELLATION=*YES).

Format

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

Operands

PARTNER = <text 1..200 with-low>
Name of the partner system from the partner list or the address of the partner system. For details on specifying partner addresses, see section “Specifying partner addresses” on page 76.

Example

Remove the remote system PARTNER1 from the partner list of the current openFT instance:

```
ftremptn partner1
```

4.26 FTSCOPY Transfer file synchronously

Note on usage

User group: FT user

Functional description

With the FTSCOPY command, you issue a synchronous request to send one or more files to the remote system or to retrieve one or more files from the remote system.

With a few exceptions, the operands are identical to those of the NCOPY command. Consequently only the syntax is described.

FTSCOPY differs from NCOPY in the following points:

- There is no local follow-up processing. The local parameters PROCESSING-ADMISSION, SUCCESS-PROCESSING and FAILURE-PROCESSING are therefore omitted.
- The general parameters PRIORITY, START and CANCEL are not used, because they do not have any significance for synchronous transfer.

Format

(part 1 of 4)

```
FTSCOPY

 TRANSFER-DIRECTION = TO-PARTNER / FROM-PARTNER
,PARTNER = <text 1..200 with-low>
,LOCAL-PARAMETER = *PARAMETERS(...)
    *PARAMETERS(...)
        FILE-NAME =*NOT-SPECIFIED /<filename 1..59> / <c-string 1..512 with-low>
       ,PASSWORD = *NONE / <alphanum-name 1..8>
       ,TRANSFER-ADMISSION = *SAME / <alphanum-name 8..32> / <x-string 15..64> /
                                   <c-string 8..32 with-low> / *PARAMETERS(...)
           *PARAMETERS(...)
               USER-IDENTIFICATION = <name 1..8>
              ,ACCOUNT = *NONE / <alphanum-name 1..40> / c-string 1..40>
              ,PASSWORD = *NONE / <alphanum-name 1..8>
       ,CODED-CHARACTER-SET = *STD / <alphanum-name 1..8>
```

(part 2 of 4)

```
,REMOTE-PARAMETER = *BS2000(...) / *MSP(...) / *ANY(...)

  *BS2000(...)
   |  FILE-NAME = *NOT-SPECIFIED / <filename 1..54> / <c-string 1..512 with-low>/
   |                 *LIBRARY-ELEMENT(...)
   |    *LIBRARY-ELEMENT(...)
   |     |  LIBRARY = *NOT-SPECIFIED / <filename 1..54>
   |     | ,ELEMENT = *NOT-SPECIFIED /  <filename 1..64 without-gen-vers>(...) /
   |     |                 <composed-name 1..64 with-under>(...) / <number 1..ffff>
   |     |     <filename>(...) / <composed-name>(...)
   |     |      |  VERSION = *STD / <text 1..24>
   |     | ,TYPE = *NOT-SPECIFIED / <alphanum-name 1..8>
   | ,PASSWORD = *NONE / <c-string 1..4> / <x-string 1..8> /
   |                  <integer -2147483648..2147483647>
   | ,TRANSFER-ADMISSION = <alphanum-name 8..32> / <x-string 15..64> /
   |                             <c-string 8..32 with-low>/ *PARAMETERS(...)
   |    *PARAMETERS(...)
   |     |  USER-IDENTIFICATION = <name 1..8>
   |     | ,ACCOUNT = *NONE / <alphanum-name 1..8>
   |     | ,PASSWORD = *NONE / <c-string 1..8> / <c-string 9..32> / <x-string 1..16>
   | ,PROCESSING-ADMISSION = *SAME / *NOT-SPECIFIED / *PARAMETERS(...)
   |    *PARAMETERS(...)
   |     |  USER-IDENTIFICATION = <alphanum-name 1..8>
   |     | ,ACCOUNT = *NONE / <alphanum-name 1..8>
   |     | ,PASSWORD = *NONE / <c-string 1..8> / <c-string 9..32> / <x-string 1..16>
   | ,SUCCESS-PROCESSING = *NONE / <c-string 1..1000 with-low>
   | ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
   | ,CODED-CHARACTER-SET = *STD / <alphanum-name 1..8>
```

(part 3 of 4)

```
*MSP(...)
   FILE-NAME = *NOT-SPECIFIED / <filename 1..59> / <c-string 1..512 with-low>
  ,PASSWORD = *NONE / <alphanum-name 1..8>
  ,TRANSFER-ADMISSION = <alphanum-name 8..32> / <x-string 15..64> / <c-string 8..32 with-low> /
                        *PARAMETERS(...)
     *PARAMETERS(...)
          USER-IDENTIFICATION = <name 1..8>
         ,ACCOUNT = *NONE<alphanum-name 1..40> / <c-string 1..40>
         ,PASSWORD = *NONE / <alphanum-name 1..8>
  ,PROCESSING-ADMISSION = *SAME / *NOT-SPECIFIED / *PARAMETERS(...)
     *PARAMETERS(...)
          USER-IDENTIFICATION = <name 1..8>
         ,ACCOUNT = *NONE<alphanum-name 1..40> / <c-string 1..40>
         ,PASSWORD = *NONE / <alphanum-name 1..8>
  ,SUCCESS-PROCESSING = *NONE /<c-string 1..1000 with-low>
  ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
  ,CODED-CHARACTER-SET = *STD / <alphanum-name 1..8>

*ANY(...)
   FILE-NAME = *NOT-SPECIFIED / <c-string 1..512 with-low> / *LIBRARY-ELEMENT(...)
     *LIBRARY-ELEMENT(...)
          LIBRARY = *NOT-SPECIFIED / <c-string 1..63 with-low>
         ,ELEMENT = *NOT-SPECIFIED / <c-string 1..64 with-low>(...)
            <c-string 1..64 with-low>(...)
                 VERSION = *NONE / *STD / <c-string 1..24 with-low>
         ,TYPE = *NONE / *NOT-SPECIFIED / <c-string 1..8 with-low>
  ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128>
  ,TRANSFER-ADMISSION = *NONE / <alphanum-name 8..32> / <x-string 15..64> /
                        <c-string 8..32 with-low> / *PARAMETERS(...)
     *PARAMETERS(...)
          USER-IDENTIFICATION = <c-string 1..67 with-low>
         ,ACCOUNT = *NONE / <c-string 1..64 with-low>
         ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128 with-low>
  ,PROCESSING-ADMISSION = *SAME / *NONE / *PARAMETERS(...)
     *PARAMETERS(...)
          USER-IDENTIFICATION = <c-string 1..67 with-low>
         ,ACCOUNT = *NONE / <c-string 1..64 with-low>
         ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128 with-low>
```

(part 4 of 4)

,**SUCC**ESS-PROCESSING = ***NONE** /<c-string 1..1000 with-low>
,**FAIL**URE-PROCESSING = ***NONE** / <c-string 1..1000 with-low>
,**CODED-CHAR**ACTER-**SET** = ***STD** / <c-string 1..8 with-low>

,**COMP**RESS = ***NONE** / ***BYTE-REPET**ITION / ***ZIP**

,**WRITE**-MODE = ***REPLACE**-FILE / ***NEW**-FILE / ***EXT**END-FILE

,**DATA**-TYPE = ***NOT-SPECIFIED** / ***CHAR**ACTER (...) / ***BIN**ARY (...) / ***USER**

***CHAR**ACTER(...)
TRANSPARENT = ***NO** / ***YES**

***BIN**ARY(...)
TRANSPARENT = ***NO** / ***YES**

,**DATA**-**ENC**RYPTION = ***NO** / ***YES** / ***ONLY-DATA-INTEGRITY**

,**REC**ORD-**SIZE** = ***NOT-SPECIFIED** /<integer 1..32756>

,**REC**ORD-**FORM**AT = ***STD** / ***FIXED** / ***VARIABLE** / ***UNDEFINED**

,**TABULATOR** = ***AUTO** / ***ON** / ***OFF**

,**TARGET-FILE-FORM**AT = ***SAME** / ***BLOCK**-ORIENTED / ***SEQ**UENTIAL(...)

***SEQ**UENTIAL(…)
RECORD-**FORM**AT = ***SAME** / ***UNDEFINED**

Operands

The meaning of the operands is the same as for asynchronous file transfer, see the operand description for NCOPY as of page 350.

Example

The file EXAMPLE is to be transferred to the remote Unix system PUX. Here, it is to be stored in the directory dir (subdirectory of the HOME directory) under the transfer admission ForUXSys. ZIP compression is to be used for transfer.

```
           FTSCOPY TRANS-DIR=TO,PARTNER=PUX,                   -
*LOCAL=*PAR(FILE-NAME=EXAMPLE),                     -
*REM=*ANY(FILE-NAME='dir/file.ux',TRANS='ForUXSys'), -
COMP=*ZIP
```

```
FTR0005 OPENFT: Request 91339. File 'EXAMPLE' transferred
```

Short form:

```
FTSCOPY TO,PUX,(EXAMPLE),*a('dir/file.ux',,'ForUXSys'),*ZIP
```

4.27 FTSHW Display remote file attributes

Note on usage

User group: FT user

Functional description

With the FTSHW command, you can display the appropriate file or files in a directory on a remote partner system.

There are three options for displaying attributes:
- List the name(s) of the file(s) in a directory
- Display a default selection of attributes returned by the partner system
- Display all attributes of a file or files in a directory, as returned by the partner system on request.

Format

```
FTSHW

 PARTNER = <text 1..200 with-low>
,FILE = *NOT-SPECIFIED / <filename 1..59> / <c-string 1..512 with-low> / <text 1..512> / *DIRECTORY(...)
    *DIRECTORY(...)
          NAME = *NOT-SPECIFIED / <c-string 1..512 with-low> / <text 1..512> / <partial-filename 2..53>
,PASSWORD = *NONE / <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>
,TRANSFER-ADMISSION = *NONE / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> /
                      *PARAMETERS(...)
    *PARAMETERS(...)
          USER-IDENTIFICATION = <name 1..8> / <c-string 1..67 with-low>
         ,ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
         ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
,INFORMATION = *STD / *ALL-ATTRIBUTES / *NAMES-ONLY
,OUTPUT = *STDERR(...) / *STDOUT(...)
    *STDERR(...) / *STDOUT(...)
          LAYOUT = *STD / *CSV
```

Operands

PARTNER = <text 1..200 with-low>
Name of the partner system as defined by the FT administrator in the partner list or the address of the partner system. For more information on address specifications, see section “Specifying partner addresses” on page 76.

FILE =
Name of the file in the remote FT partner system.

FILE = *NOT-SPECIFIED
The name of the file is known to the remote system because it has already been completely defined in the addressed FTAC admission profile, for instance.

FILE = <filename 1..59> / <c-string 1..512 with-low> / <text 1..512>
Name of the file in the remote system. The file name must be specified in the syntax of the remote system and conform to the conventions of the remote system.
If the file name is specified with an unaccessible 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 = <u>*NOT-SPECIFIED</u>
The name of the directory is known to the remote system because it has already been completely defined in the addressed FTAC admission profile, for instance.

NAME = <c-string 1..512 with-low> / <text 1..512> / <partial-filename 2..53>
Name of the directory in the remote FT partner system. The directory name must be specified in the syntax of the remote system and must conform to the conventions of the remote system
If the remote system is a BS2000 or z/OS 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.

i 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 Windows or Unix system, no password is required.

PASSWORD = <u>*NONE</u>
Access is possible without a password.

PASSWORD = <integer -2147483648..2147483647> / <c-string 1..64 with-low> / <x-string 1..128>
Password that allows the user to access the file in the remote system. The password must be specified in the syntax of the remote system and must conform to the conventions of the remote system.

TRANSFER-ADMISSION =
Transfer admission in the remote system for the file management request.

TRANSFER-ADMISSION = <u>*NONE</u>
The remote system does not require or recognize any user authorization.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
The transfer admission for the remote system can only be defined in an FT profile if FTAC functionality has been installed on the remote system. In this case, only the TRANSFER-ADMISSION defined in the FT profile is specified. The alphanumeric entry is converted internally to lowercase characters.

TRANSFER-ADMISSION = *PARAMETERS(...)
Identification, account number and password of the user in the remote system. The operands in brackets can also be used as positional operands without their keywords.

USER-IDENTIFICATION = <name 1..8> / <c-string 1..64 with-low>
Identification of the user in the remote system. The identification must be specified in the syntax of the remote system and must conform to the conventions of the remote system.

ACCOUNT = *NONE / <c-string 1..64 with-low> / <text 1..64>
Account number of the user in the remote system. The account number must be specified in the syntax of the remote system and must conform to the conventions of the remote system.

PASSWORD =
Password that allows the user to access the remote system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD =
<c-string 1..64 with-low> / <x-string 1..128> / <alphanum-name 1..19>
Password that allows the user to access the remote system. The password must be specified in the syntax of the remote system, must conform to the conventions of the remote system, and must be recognized by the remote system.

INFORMATION =
Amount of information required. The amount of information is also dependent on the amount of information supplied by the partner. Therefore, only some attributes may be displayed, even if you requested full information.

INFORMATION = *STD
The default range of information is output.

INFORMATION = *ALL-ATTRIBUTES
All available information is requested on the file in the partner system. However, only attribute values returned by the partner system can be displayed.

INFORMATION = *NAMES-ONLY
Only the names of the directory files or the name of the file is listed.

OUTPUT =
Output medium.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or SYSERR, if this DDNAME is defined.

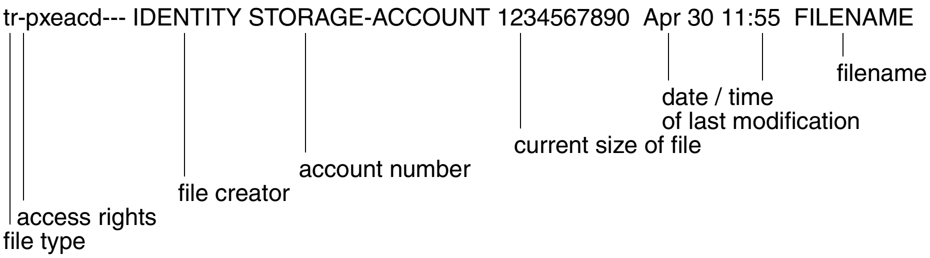
OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

LAYOUT = *STD
Output is put into a user-friendly form for reading.

LAYOUT = *CSV
Output is in **C**haracter **S**eparated **V**alues format. This is a special, tabular format, widely used in the PC world, in which the individual fields are separated by a semicolon “;” (see page 426).

4.27.1 Output of FTSHW

Default output



```
tr-pxeacd--- IDENTITY STORAGE-ACCOUNT 1234567890  Apr 30 11:55  FILENAME
```

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
```

4.27.2 Description of output fields of FTSHW

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.

FILESIZE – 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 2017).

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).

Examples

1. You want to view the properties of the PS file *dat1* with a variable record format of the length 255 on a z/OS system with the partner name *ZOS2PART*:

   To do this, you can issue the following command under TSO:

```
FTSHW ZOS2PART,FILE-NAME=DAT1,TRANS-ADM=(USERID,ACCT,PASSWORD),INF=*ALL
```

```
FILENAME=DAT1
 CRE   OPFT000
 MOD   DATE=Mar 25 00:00
 RECORD-FORMAT=v RECORD-SIZE=255        FILE-AVAILABILITY=i
 ACCESS-RIGHTS=r-pxeacd---   FILESIZE=8192
```

2. You want to view the properties of the file *dat2* on a BS2000 system with the partner name *BS2PART*:

```
FTSHW BS2PART,FILE-NAME=DAT2,TRANS-ADM=(USERID,ACCT,PASSWORD),INF=*ALL
```

```
FILENAME=DAT2
 CRE   OPENFT
 MOD   DATE=Mar 10 2012
 CHARACTERSET=g
 RECORD-FORMAT=f RECORD-SIZE=80
 ACCESS-RIGHTS=r-pxeacd---   FILESIZE=2048
```

3. You want to view the properties of the file *dat3* on the Windows system with the partner name *WINPC* and to do so, specify the FTAC transfer admission *FTACTRANSADM*:

   ```
   FTSHW WINPC,FILE-NAME=DAT3,TRANS-ADM=('FTACTRANSADM')
   ```

   ```
   *r----a----- FT:mueller    27185709  Sep 17 2016  DAT3
   ```

   The command for detailed output is as follows:

   ```
   FTSHW WINPC,FILE-NAME=DAT3,TRANS-ADM=('FTACTRANSADM'),INF=*ALL
   ```

   ```
   FILENAME=DAT3
    CRE   FT:mueller
    MOD   DATE=Sep 17 2016
    RECORD-FORMAT=u FILE-AVAILABILITY=i
    ACCESS-RIGHTS=r----a-----   FILESIZE=27185709
   ```

4. You want to view the properties and members of the PO library *OPFT000.POBIB.CLIST* on a z/OS system with the user ID *opft000* and the partner name *ZOS2PART*:

   ```
   FTSHW ZOS2PART,*DIR=('OPFT000.POBIB.CLIST')
                 ,TRANS-ADM=(OPFT000,ACCT,PASSWORD)

    *r-pxeacd--- OPFT000              0  Jul 16 00:00 MEMBER1
    *r-pxeacd--- OPFT000              0  Jul 16 00:00 MEMBER2
    *r-pxeacd--- OPFT000              0  Jul 16 00:00 MEMBER3
   ```

   The command for detailed output is as follows:

   ```
   FTSHW ZOS2PART,*DIR=('OPFT000.POBIB.CLIST')
                 ,TRANS-ADM=(OPFT000,ACCT,PASSWORD),INF=*ALL

    FILENAME=MEMBER1
    CRE   OPFT000
    MOD   DATE=Jul 16 00:00
    RECORD-FORMAT=f RECORD-SIZE=80         FILE-AVAILABILITY=i
    ACCESS-RIGHTS=r-pxeacd---   FILESIZE=0
    FILENAME=MEMBER2
    CRE   OPFT000
    MOD   DATE=Jul 16 00:00
    RECORD-FORMAT=f RECORD-SIZE=80         FILE-AVAILABILITY=i
    ACCESS-RIGHTS=r-pxeacd---   FILESIZE=0
    FILENAME=MEMBER3
    CRE   OPFT000
    MOD   DATE=Jul 16 00:00
    RECORD-FORMAT=f RECORD-SIZE=80         FILE-AVAILABILITY=i
    ACCESS-RIGHTS=r-pxeacd---   FILESIZE=0
   ```

4.28 FTSHWADS Display admission sets

Note on usage

User group: FTAC user and FTAC administrator

Prerequisite for using this command is the use of openFT-AC.

Functional description

You use the FTSHWADS command to display admission sets. You can output the following information on either SYSTSPRT or SYSPRINT:

- if the admission set is privileged (if so, then you are the FTAC administrator).
- if a password is required to use FTAC commands on this user ID. The password itself is not displayed.
- the limiting values for accessible security levels which have been set by the owner of this user ID.
- the limiting values for accessible security levels which have been pre-set by the FTAC administrator.

Format

```
FTSWHADS
-----------------------------------------------------------
 USER-IDENTIFICATION = *OWN / *ALL / *STD / <name 1..8>
,SELECT-PARAMETER = *ALL
,OUTPUT = *STDERR(...) / *STDOUT(...)
   *STDERR(...) / *STDOUT(...)
      |   LAYOUT = *STD / *CSV
```

Operands

USER-IDENTIFICATION =
User ID whose admission set you wish to view. FTAC users can only obtain information about their own admission set and the standard admission set. The FTAC administrator can obtain information about any admission set.

USER-IDENTIFICATION = *OWN
FTAC outputs your own user ID’s admission set.

USER-IDENTIFICATION = *ALL
FTAC outputs the standard admission set and the admission set of your own user ID. For the FTAC administrator, all admission sets are output which differ from the standard admission set.

USER-IDENTIFICATION = *STD
FTAC only outputs the standard admission set.

USER-IDENTIFICATION = <name 1..8>
FTAC outputs the admission set that belong to the user ID specified. The FTAC user can only enter his/her own user ID here. The FTAC administrator can enter any user ID.

SELECT-PARAMETER = *ALL
This parameter is reserved for future extensions and has no effect in the current version.

OUTPUT =
Output medium for the information requested.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or to SYSERR if this DDNAME is defined. If the command is called with ftexec from a Unix or Windows system, ftexec sends the output to stderr.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT. If the command is called with ftexec from a Unix or Windows system, ftexec sends the output to stdout.

LAYOUT = *STD
Output is formatted using a standard layout that can be easily read by the user.

LAYOUT = *CSV
Output is supplied in CSV (**C**haracter **S**eparated **V**alues) format. This is a widely used tabular format, especially in the PC environment, in which individual fields are separated by a delimiter, which is usually a semicolon “;“ (see page 428).

4.28.1 Output of FTSHWADS

Example

Jack John, the FTAC administrator of the Dack Bank, wants to obtain information about the admission sets in his system. He enters the following command:

```
FTSHWADS␣USER-IDENTIFICATION=*ALL
```

Short form:

```
FTSHWADS␣*ALL
```

He receives the following output:

```
              MAX. USER LEVELS              MAX. ADM LEVELS          ATTR
USER-ID  OBS  OBR  IBS  IBR  IBP  IBF  OBS  OBR  IBS  IBR  IBP  IBF
*STD      10   10   10   10    0    0   10   10   10   10    0    0
JACK     100  100    0    0    0*   0* 100  100    0    0    0*   0* PRIV
GRACE     50   50   10*  50   50   50   50   50   50   50   50   50  PW
DANIEL     0   10    0    0    0    0   10   10    0    0    0    0  PW
STEVEN    50  100    0   10*   0    0   50  100   10   50    0    0
```

These can be explained as follows:

The user ID of each admission set is in the column USER-ID. In this example, there is a standard admission set as well as admission sets for the user IDs JACK, GRACE, DANIEL and STEVEN.
The FTAC user sees the standard 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). 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 standard 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 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 standard admission set and will change if any modifications are made to the standard admission set.

4.29 FTSHWENV Display saved admission profiles and sets

Note on usage

User group: FTAC administrator

openFT-AC must be installed to use this command.

Functional description

The FTAC administrator can use the command FTSHWENV to view admission profiles and sets which have been written in an export file using the command FTEXPENV (see page 159). This function is particularly useful before the importing of the admission profiles and sets (see page 163).

Format

```
FTSHWENV

 FROM-FILE = <filename 1..46>
,USER-IDENTIFICATION = *ALL / list-poss(100): <name 1..8>
,SELECT-PARAMETER = *ALL / *PARAMETERS(...)
   *PARAMETERS(...)
      |  PROFILE-NAME = *ALL / *NONE / *STD / list-poss(100): <alphanum-name 1..8>
      | ,ADMISSION-SET = *YES / *NO
,INFORMATION = *ONLY-NAMES / *ALL
,OUTPUT = *STDERR(...) / *STDOUT(...)
   *STDERR(...) / *STDOUT(...)
      |  LAYOUT = *STD / *CSV
```

Operands

FROM-FILE = <filename 1..46>
Name of the file (not a temporary file) from which the admission profiles and sets are to be displayed. If the file contains invalid data or access to the file is unsuccessful, the command is rejected with the message FTC0103.

USER-IDENTIFICATION =
User ID whose admission profiles and sets are to be displayed.

USER-IDENTIFICATION = *ALL
The admission profiles and sets of all users are to be displayed.

USER-IDENTIFICATION = list-poss(100): <name 1..8>
The admission profiles and sets of the user IDs specified (maximum 100) are to be displayed.

SELECT-PARAMETER =
Specifies whether only admission profiles, only admission sets or both are to be displayed. For the admission profiles, you can specify which ones are to be displayed.

SELECT-PARAMETER = *ALL
All admission profiles and sets associated with the user ID specified under USER-IDENTIFICATION are to be output on file.

SELECT-PARAMETER = *PARAMETERS(...)
Specifies which of the admission sets associated with the USER-IDENTIFICATION are to be specified.

> **PROFILE-NAME = *ALL**
> All admission profiles are displayed.
>
> **PROFILE-NAME = *NONE**
> No admission profiles are displayed.
>
> **PROFILE-NAME = *STD**
> Displays the standard admission profile.
>
> **PROFILE-NAME = list-poss(100): <alphanum-name 1..8>**
> Only the specified profiles are displayed (maximum 100).
>
> **ADMISSION-SET = *YES**
> All admission sets are displayed.
>
> **ADMISSION-SET = *NO**
> No admission sets are displayed.

INFORMATION =
Scope of the information to be displayed.

INFORMATION = *ONLY-NAMES
Only the names of the admission profiles are to be displayed.

INFORMATION = *ALL
The entire contents of the admission profiles, excluding any passwords and transfer admissions, are displayed.

OUTPUT =
Output medium.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or to SYSERR if this DDNAME is defined. If the command is called with ftexec from a Unix or Windows system, ftexec sends the output to stderr.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT. If the command is called with ftexec from a Unix or Windows system, ftexec sends the output to stdout.

LAYOUT = *STD
Output is formatted using a standard layout that can be easily read by the user

LAYOUT = *CSV
Output is supplied in CSV (**C**haracter **S**eparated **V**alues) format. This is a widely used tabular format, especially in the PC environment, in which individual fields are separated by a delimiter, which is usually a semicolon “;“ (see page 430).

Example

The FTAC administrator Jack John backs up the admission set and the admission profiles of the user ID STEVEN in the file STEVEN.FTAC.BKUP.

```
FTEXPENV␣TO-FILE=STEVEN.FTAC.BKUP,USER-IDENTIFICATION=STEVEN
```

A possible short form of this command would be:

```
FTEXPENV␣STEVEN.FTAC.BKUP,STEVEN
```

As a conscientious FTAC administrator, Jack checks if the desired back-up is in the file STEVEN.FTAC.BKUP

```
FTSHWENV␣FROM-FILE=STEVEN.FTAC.BKUP
```

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
 STEVEN     1    1    0    1    0    0     1    1    0    0    0    0
 OWNER     NAME
 STEVEN   *UMSAWARE
```

USER-ID and OWNER can be used to determine the user ID with which the admission sets and profiles defined under NAME are associated.

In addition, the maximum security levels set for each user are displayed, as in the command FTSHWADS. An explanation of these entries can be found in the section for this command (page 268).

4.30 FTSHWINS
Display an openFT instance

Note on usage

User group: FT user

This command must be called under TSO.

The command only works if openFT has been started as a subsystem. If openFT has been started as a batch job, the message `INSTANCES MGMT NOT AVAILABLE` is issued.

Functional description

With the command FTSHWINS you can display information regarding openFT instances.

Format

FTSHWINS
[***ALL**]

Operands

INSTANCES = *ALL
Outputs a list of all available instances.

FTSHWINS with no operand specified displays the currently set instance.

Example

```
FTSHWINS *ALL
NAME   FTID   PORT   IP-ADDRESS          SERVICES     TIME    DATE
----------------------------------------------------------------------
STD    PBFT2  1100   123.45.67.89              44    7:51:06  2017-145
FTBLA  KUHEM  3040   123.45.67.99               3   15:16:33  2017-146
```

4.31 FTSHWKEY Show properties of RSA keys

Note on usage

User group: FT administrator

This command must be called under TSO.

Functional description

You can use the FTSHWKEY command to output the properties of RSA keys. You can display the RSA keys of your own instance as well as the RSA keys of partners.

Format

```
FTSHWKEY

 SELECT = *ALL / *OWN / *PARAMETERS (...)

   *PARAMETERS(...)
     |   PARTNER-NAME = *ALL / <name 1..8>
     |   ,EXPIRATION-DATE = *NOT-SPECIFIED / *NONE / *EXCEEDED / *UNTIL(DATE = <date 8..10>) /
     |              *WITHIN(DAYS = <integer 1..1000>)

,OUTPUT = *STDERR(...) / *STDOUT(...)

   *STDERR(...) / *STDOUT(...)
     |   LAYOUT = *STD / *CSV
```

Operands

SELECT =
Selects which keys are to be displayed.

SELECT = *ALL
Displays the keys of your own instance and the installed keys of all the partner systems.

SELECT = *OWN
Displays the keys of your own instance.

SELECT = *PARAMETERS(...)
Specifies selection criteria for the keys which are to be displayed.

PARTNER-NAME =
Partner whose key is to be displayed.

PARTNER-NAME = *ALL
Displays the installed keys of all partners.

PARTNER-NAME = <name 1..8>
Name of the partner whose key is to be displayed.

EXPIRATION-DATE =
Selects keys on the basis of their expiration date.

EXPIRATION-DATE = *NOT-SPECIFIED
The keys of the partners are displayed irrespective of their expiration date.

EXPIRATION-DATE = *NONE
Displays all partner keys that do not have an expiration date.

EXPIRATION-DATE = *EXCEEDED
Displays all partner keys that have already expired.

EXPIRATION-DATE = *UNTIL(...)
Displays all partner keys that will become invalid by a particular date.

DATE=<date 8...10>
Date in the format *yyyy-mm-dd* or *yy-mm-dd*, e.g. 2018-03-31 or 18-03-31 for March 31, 2018, by which date the keys will become invalid. The time on the specified day is 00:00 local time.

EXPIRATION-DATE = *WITHIN(...)
Displays all partner keys that will expire within the specified number of days.

DAYS = <integer 1...1000>
Number of days within which the keys will become invalid. The time on the last day of the period is 00:00 local time.

OUTPUT =
Output medium for the requested information.

OUTPUT = *STDERR(...)
Output is written to SYSTSPRT or to SYSERR if this DDNAME is defined. When the command is called from a Unix or Windows system using ftexec, ftexec writes the output there to stderr.

OUTPUT = *STDOUT(...)
Output is written to SYSPRINT. When the command is called from a Unix or Windows system using ftexec, ftexec writes the output there to stdout.

LAYOUT = *STD
Output takes place in a format which is easy for the user to read.

LAYOUT = *CSV
Output takes place in **C**haracter **S**eparated **V**alues format. This is a table-type format which is widely used, especially in the PC environment, and in which the individual fields are separated by a semicolon “;“ (see page 431).

Example

```
FTSHWKEY
CRE-DATE   EXP-DATE   KEY-LEN KEY-REF   AUTHL  PARTNER IDENTIFICATION
2016-08-31            768     2         2
2016-08-31            1024    2         2
2016-08-31            2048    2         2
2016-10-31            1024    3         2
2016-11-29            2048    4         2
2016-08-28 2018-03-31 2048    7         2      MYOWN   MYOWNID.DOMAIN.NET
2015-03-12 EXPIRED    768     12        2      PC17QD  PC17QD.DOMAIN.NET
2016-06-14            1024    1036      1      PC27ABC PC27ABC.DOMAIN.NET
```

Explanation:

CRE-DATE
Date on which the key was generated.

EXP-DATE
Date on which the key expires. The time on the specified day is 00:00 local time. EXPIRED means that the key has already expired.

If there is no specification here then there is no expiration date.

KEY-LEN
Key length in bits: 768, 1024 or 2048

KEY-REF
Key reference

AUTHL Authentication level: 1 or 2

PARTNER
Partner’s name. This field is left empty for keys belonging to your own instance.

IDENTIFICATION
Partner’s instance ID. This field is left empty for keys belonging to your own instance.

4.32 FTSHWLOG Display log records and offline log files

Note on usage

User group: FT user, FT administrator and FTAC administrator

Functional description

With the FTSHWLOG command, you can obtain information on all FT requests logged by openFT. An important prerequisite is that the FT administrator has switched on the FT logging function. The logging records are marked as FT or FTAC or ADM, enabling you to identify the type of logging record.

FTSHWLOG also enables the name of the current log file and the names of the offline log files to be displayed.

FT logging

The FT user can view all log records which relate to his/her user ID. The FT administrator can display all the FT log records in the system.

If no options are specified, openFT outputs the most recent log record. When requested, openFT outputs all the log records which correspond to the selection criterion defined in the command.

Command execution may take several minutes, depending on the size of the log file!

There are three types of output: short output and long output and CSV format.

FTAC logging

With FTAC functionality, FTSHWLOG can be used to display the FTAC log records. The FT user can view all FTAC log records, of which he/she is the owner. FT and FTAC administrators may view all FTAC log records.

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.

Note

> It is not necessary for FT and FTAC administrators to cooperate in order to sure that logging data is archived in full.

A precise description of output can be found starting on page 295.

ADM logging

If your openFT instance is administered via a remote administration server or if you administer other instances yourself using FTADM, ADM log records are written (assuming that the appropriate logging settings have been made). You can also view these log records.

Format

(part 1 of 2)

```
FTSHWLOG

SELECT = *OWN / *ALL / *PARAMETERS(...)

  *PARAMETERS(...)
      LOGGING-ID = *ALL / <alphanum-name 1..12> / *INTERVAL(...)
        *INTERVAL(...)
            FROM = 1 / <alphanum-name 1..12>
           ,TO = *HIGHEST-EXISTING / <alphanum-name 1..12>
     ,OWNER-IDENTIFICATION = *OWN / *ALL / <name 1..8>
     ,CREATION-TIME = *INTERVAL(...) / *DAYS(...)
        *INTERVAL(...)
            FROM = 2000-01-01(...) / <date 8..10>(...)
              <date 8..10>(...)
                  TIME = 00:00 / <time 1..8>
           ,TO = *TOMORROW(...) / *TODAY(...) / <date 8..10>(...)
              <date 8..10>(...)
                  TIME = 00:00 / <time 1..8>
        *DAYS(...)
            NUMBER = <integer 1..1000>
     ,RECORD-TYPE = *ALL / *PARAMETERS(...)
        *PARAMETERS(...)
            FT = *TRANSFER-FILE / *NONE / list-poss(1): *TRANSFER-FILE
           ,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 / *DELETE-DIRECTORY /
                   *MODIFY-DIRECTORY / *LOGIN
           ,ADM = *ADMINISTRATION / *NONE / list-poss(1): *ADMINISTRATION
     ,INITIATOR = (*LOCAL, *REMOTE) / list-poss(2): *LOCAL / *REMOTE
     ,PARTNER = *ALL / <text 1..200 with-low>
     ,FILE-NAME = *ALL / <filename 1..59> / <filename-prefix 2..50> /
              <c-string 1..512 with-low> / *DIRECTORY(...)
        *DIRECTORY(...)
            NAME = *ALL / <partial-filename 2..50> / <c-string 1..512 with-low>
     ,REASON-CODE = *ALL / *FAILURE / <text 1..4>
```

(part 2 of 2)

```
          ,ROUTING-INFO = *ALL / <text 1..200 with-low>
          ,TRANSFER-ID = *ALL / <integer 1.. 2147483647>
          ,GLOBAL-REQUEST-ID = *ALL / <alphanum-name 1..10>
          ,LOGGING-FILE = *CURRENT / <filename 1..42> /  *ACTIVE-AT(...)
             *ACTIVE-AT(...)
                 DATE = <date 8..10>
                 ,TIME = 00:00 / <time 1..8>
          ,PREVIOUS-FILES = *STD / <integer 0..3>
,NUMBER = 1 / *ALL / <integer 1..99999999> / *POLLING(...)
   *POLLING(...)
          INTERVAL = 1 / <integer 1..600>
          ,NUMBER = *UNLIMITED / <integer 1..3600>
,INFORMATION = *STD / *ALL / *LOGGING-FILES
,OUTPUT = *STDERR(...) / *STDOUT(...)
   *STDERR(...) / *STDOUT(...)
          LAYOUT = *STD / *CSV
```

Operands

SELECT =
Selects a group of logging records.

SELECT = *OWN
Selects logging records under the user’s own login.

SELECT = *ALL
Displays all users’ logging records to the administrator. 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
The logging records of all user IDs are displayed. The FT or FTAC administrator can thus display the FT logging records of any user ID.
Normal FT users receive information only on the logging records of their own ID even if *ALL is specified.

OWNER-IDENTIFICATION = <name 1..8>
Any user ID whose logging records should be displayed.
FT users may only specify their own ID.

CREATION-TIME =
The range of the logging records to be output, selected by their date or time of creation.

CREATION-TIME = *INTERVAL(...)
The range is specified as a time interval using the date and/or time.

FROM = 2000-01-01(...) / <date 8..10>(...)
Date in the format *yyyy-mm-dd* or *yy-mm-dd*, e.g. 2016-08-18 or 16-08-18 for 18 August, 2016. 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. 2016-08-18 or 16-08-18 for 18 August, 2016. 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 records 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 = <u>*ADMINISTRATION</u> / *NONE / list-poss(1): *ADMINISTRATION
Specifies whether ADM log records are output.

ADM = <u>*ADMINISTRATION</u>
ADM log records are output. For further details, refer to the manual "openFT (z/OS) - Installation and Operation".

ADM = *NONE
No ADM log records are output.

INITIATOR =
Logging records according to the initiator.

INITIATOR = (<u>*LOCAL,*REMOTE</u>)
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 “Specifying partner addresses” on page 76.

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 question mark may not, however, be in first place.

FILE-NAME =
File name.

FILE-NAME = *ALL
The file name is not a selection criterion.

FILE-NAME = <filename 1..59> / <c-string 1..512 with-low>
Fully qualified name of the files for which you wish to view the logging records.

FILE-NAME = <filename-prefix 2..50>
Partially qualified name of the files for which you want to view the logging records.

Examples

- If you specify TOOLS as the beginning of the filename, all logging records containing the filename TOOLS.CLIST, TOOLS.CNTL or TOOLS.CLIST(MEMBER01) will be displayed.
- If you specify TOOLS.CLIST/ as the beginning of the filename, all logging records containing the filename TOOLS.CLIST(MEMBER01), TOOLS.CLIST(MEMBER02), etc. are displayed.

FILE-NAME = *DIRECTORY(...)
Name of the directory.

***DIRECTORY(...)**
Here you specify the directory in the same format as used on the partner computer in one of the openFT user commands CREATE-/MODIFY-/DELETE-REMOTE-DIR or FTSHW (see page 260).

NAME = *ALL
The directory is not a selection criterion

NAME = <partial-filename 2..50> / <c-string 1..512 with-low>
Name of the directory.

Example

If you specify FILE=*DIR(NAME=ABC.) here, and not FILE=ABC., only those logging records are displayed that contain ABC (as the name of a file directory which were accessed from a remote system with the file management command in order to display an z/OS file directory).

REASON-CODE =
Selection by the reason code of the logging records.

REASON-CODE = *ALL
The reason code is not a selection criterion; all records are output.

REASON-CODE = *FAILURE
All logging records with error codes are output.

REASON-CODE = <text 1..4>
Logging records to be output by the error codes. Leading zeros can be omitted (e.g. 14 for FTR0014).

ROUTING-INFO = *ALL / <text 1..200 with-low>
Selects the ADM log records on the basis of the routing information. The routing information describes the administered instance in the case of remote administration requests issued locally.

ROUTING-INFO = *ALL
The routing information is not used as a selection criterion.

ROUTING-INFO = <text 1..200 with-low>
Routing information for which the ADM log records are to be output.

TRANSFER-ID =
Selection on the basis of the request ID.

TRANSFER-ID = *ALL
The request ID is not used as a selection criterion.

TRANSFER-ID = <integer 1..2147483647>
Only outputs log records for the specified request ID.

GLOBAL-REQUEST-ID = *ALL / <alphanum-name 1..10>
Selects the log records on the basis of the global request ID.

GLOBAL-REQUEST-ID = *ALL
The global request identification is not a search criterion.

GLOBAL-REQUEST-ID = <alphanum-name 1..10>
Outputs log records for the specified global request identification. The global request identification is relevant only for inbound requests of openFTpartners. It is assigned by the initiator of the request (transfer ID) and transferred to the local system.

LOGGING-FILE =
Selects the log file whose logging records or name are to be output. This means that you can also view offline log records.

LOGGING-FILE = *CURRENT
The current log file is selected.

LOGGING-FILE = <filename 1..42>
Specifies the name of the log file which is to be searched. If you specify a value > 0 in the PREVIOUS-FILES operand, further, older offline log files are also searched (if any exist).

LOGGING-FILE = *ACTIVE-AT(...)
Selects the log file using its creation time (local time). The log file which was created on or before the specified time is selected. If more than one log file matches the specified time, the most recent of these log files is selected. If you specify a value > 0 in the PREVIOUS-FILES operand, further, older offline log files are also searched (if any exist).

DATE = <date 8..10>
Creation date in the format *yyyy-mm-dd* or *yy-mm-dd*, e.g. 2017-01-31 or 17-01-31 for Januray 31, 2017.

TIME = 00:00 / <time 1..8>
Creation time on the date specified with DATE. You specify the time in the format *hh:mm:ss*, e.g. 14:30:10.

PREVIOUS-FILES =
Specifies the number of preceding offline log files that are to be selected in addition to the current file or the file specified with LOGGING-FILE.

PREVIOUS-FILES = *STD
The effect depends on the specification in the INFORMATION operand:

– INFORMATION = *STD (default value) or *ALL: The current lo file or the log file specified with LOGGING-FILE is searched for log records.

– INFORMATION = *LOGGING-FILES: The names of all log files are output (maximum of 1024).

PREVIOUS-FILES = <0..3>
Specifies the number of preceding offline log files (0 to 3) that are to be searched in addition to the current file or the file specified with LOGGING-FILE or whose names are to be output.

NUMBER =
Maximum number of log records or polling intervals for outputting log records.

NUMBER = 1 / <integer 1..99999999>
The maximum number of logging records that are to be displayed. The default value is 1.

NUMBER = *ALL
All logging records are displayed.

NUMBER = *POLLING(...)
Specifies that the output of log records will be repeated at regular intervals. You can define the polling interval and the number of repetitions. Irrespective of the specifications in INTERVAL and NUMBER, the most recent log record which exists is always output first.

> **INTERVAL = 1 / <integer 1...600>**
> Polling interval in seconds. On each repetition, all the new log records are filtered in accordance with the specified selection criteria and the detected records are output. By default the output is repeated every second.
>
> **NUMBER =**
> Number of repetitions.
>
> **NUMBER = *UNLIMITED**
> The output is repeated without restriction. You can, for example, cancel the output using the key combination PA1 and RESET.
>
> **NUMBER = <integer 1..3600>**
> Specifies the number of repetitions.

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 292).

INFORMATION = *ALL
The logging records are displayed in a detailed format (see page 295).

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(…)
- PREVIOUS-FILES in SELECT=*PARAMETERS(…)
- OUTPUT

OUTPUT =
Output medium.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or to SYSERR if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

LAYOUT = *STD
Output is formatted using a standard layout that can be easily read by the user.

LAYOUT = *CSV
Output is supplied in CSV (**C**haracter **S**eparated **V**alues) format. This is a widely used tabular format, especially in the PC environment, in which individual fields are separated by a delimiter, which is usually a semicolon “;“ (see page 432).

4.32.1 Description of short output of FTSHWLOG

Short output form of FT logging records (example)

```
FTSHWLOG NUMBER=2

 TYP LOGG-ID  TIME     RC    PARTNER  INITIATOR INIT USER-ADM FILENAME
 2017-04-22
 T       5333 14:18:24 2169 <G133H301 FT2V292        FT2V292  TEST2
 T       5284 14:08:12 0000 >G133H301 FT2V292        FT2V292  TEST1
```

Short output format for ADM log records (examples)

ADM log for a remote administration request that has been issued locally:

```
FTSHWLOG NUMBER=1

 TYP LOGG-ID TIME     RC    PARTNER  INITIATOR INIT USER-ADM FILENAME
 2017-04-03
 A     3555 17:15:11 0000 <ftadm:/* FTV520         FTV520
```

ADM log record on the administered openFT instance:

```
FTSHWLOG NUMBER=1

 TYP LOGG-ID TIME     RC    PARTNER  INITIATOR INIT USER-ADM FILENAME
 2017-04-03
 A     9006 11:32:51 0000 >ftadm:/* *REMOTE        ftadmin
```

Explanation

Not all values are displayed for all log record types and request types. The table below also describes values that can occur only in ADM log records.

Name	Explanation
TYP (column 1)	Specifies if it is an FT or FTAC or ADM or FTP log record. T indicates the FT logging record, C indicates the FTAC logging record, A indicates the ADM logging record.
TYP (columns 2-3)	Definition of FT function: ␣ transfer file V transfer file and delete send file (only inbound possible) A read file attributes D delete file C create file M modify file attributes R read directory CD create director MD modify directory DD delete directory L login (inbound FTP access)
LOGG-ID	Number of the log record (up to twelve digits)
TIME	Time when the logging record was written
RC	Reason Code. Indicates if a request was successfully executed, or if not, why it was rejected or terminated. If an FT request is rejected for “FTAC reasons” (e.g. 0014), the exact reason behind the termination can be found in the FTAC logging record of the system that rejected the request. Further information on the reason code can be obtained using the FTHELP xxxx command.

Name	Explanation	
PARTNER	Provides information about the partner system. The output in the case of named partners consists of the symbolic name, and in the case of dynamic partners of the address (up to 8 characters; if the address is longer, the last character is an'*'). The partner system is prefixed by an identifier from which you can determine the request direction.	
	>	The request direction is to the partner system. This direction is specified for a – send request, i.e. the data is transferred to the partner – request to view remote file attributes – request to view remote directories
	<	The request direction is to the local system. This direction is specified for a – receive request, i.e.the data is transferred to the local system – request to modify remote file attributes – request to delete remote files
INITIATOR	Initiator (user ID) in the case of requests issued locally issued; if initiative is from remote system: *REMOTE	
INIT	The field is always empty in z/OS and is only output for reasons of compatibility.	
USER-ADM	User ID in the local system used by the requests	
FILENAME	Filename resp. preprocessing or postprocessing in the local system. In the case of ADM logging records, this field is empty. For security reasons, only the first 32 characters (or 42 characters in the case of FTEXECSV preprocessing operations) of a preprocessing or postprocessing command are taken over into the logging record. By arranging the call parameters accordingly or by inserting spaces, you can influence the command parameters that are not to appear in the logging record. FTEXECSV is the reaction to an ftexec command issued in a remote Windows or Unix system.	

4.32.2 Description of long output of FTSHWLOG

Long output form outbound (example)

```
LOGGING-ID  = 9479      RC      = 0000        TIME     = 2016-07-11 14:31:29
   TRANS    = TO        REC-TYPE= FT          FUNCTION = TRANSFER-FILE
   PROFILE  =           PCMD    = NONE        STARTTIME= 2016-07-11 14:31:29
   TRANS-ID = 67052     WRITE   = REPLACE     REQUESTED= 2016-07-11 14:31:28
   TRANSFER =         1 kB                    CCS-NAME = IBM1047
   SEC-OPTS = ENCR+DICHK, RSA-2048 / AES-256
   INITIATOR= OPFTUID
   USER-ADM = OPFTUID
   PARTNER  = BS2PART
   FILENAME = FILE.TEST
   FNC-MODE = *TRANSPARENT
```

```
LOGGING-ID  = 9478      RC      = 0000        TIME     = 2016-07-11 14:31:28
   TRANS    = TO        REC-TYPE= FTAC        FUNCTION = TRANSFER-FILE
   PROFILE  =           PRIV    =
   INITIATOR= OPFTUID
   USER-ADM = OPFTUID
   PARTNER  = BS2PART
   FILENAME = FILE.TEST
```

Long output form inbound (example)

```
LOGGING-ID  = 9473      RC      = 0000        TIME     = 2016-07-11 14:25:00
   TRANS    = FROM      REC-TYPE= FT          FUNCTION = TRANSFER-FILE
   PROFILE  =           PCMD    = NONE        STARTTIME= 2016-07-11 14:24:59
   TRANS-ID = 67046     WRITE   = REPLACE     STORETIME= 2016-07-11 14:25:00
   TRANSFER =         1 kB                    CCS-NAME = IBM1047
   SEC-OPTS = ENCR+DICHK+DENCR+DDICHK+LAUTH2+RAUTH2, RSA-1024 / AES-256
   INITIATOR= *REMOTE                         GLOB-ID  = 66279
   USER-ADM = OPFTUID
   PARTNER  = BS2PART
   PTNR-ADDR= BS2PART
   FILENAME = TEST1
   FNC-MODE = *TRANSPARENT
```

```
LOGGING-ID  = 9472      RC      = 0000        TIME     = 2016-07-11 14:24:59
   TRANS    = FROM      REC-TYPE= FTAC        FUNCTION = TRANSFER-FILE
   PROFILE  = PROFIL1   PRIV    = NO
   INITIATOR= *REMOTE                         GLOB-ID  = 66279
   USER-ADM = OPFTUID
   PARTNER  = BS2PART
   PTNR-ADDR= BS2PART
   FILENAME = TEST1
```

Long output format for an ADM log record (example)

```
LOGGING-ID  = 299120    RC      = 0000        TIME     = 2016-08-29 08:55:12
   TRANS    = TO        REC-TYPE= ADM         FUNCTION = REM-ADMIN
   TRANS-ID = 156730    PROFILE = Profil06
   SEC-OPTS = ENCR+DICHK, RSA-2048 / AES-256
   INITIATOR= *REMOTE                         GLOB-ID  = 17232
   USER-ADM = FTADMIN1
   PARTNER  = REMADMIN
   ADM-CMD  = FTSHWLOG
   ADMIN-ID =
   ROUTING  =
```

Explanation of long output form (column-wise)

The table below also describes fields and values that can only occur in ADM log records.

Name	Explanation	
LOGGING-ID	Number of the log record (up to twelve digits)	
TRANS	Transfer direction:	
	TO	The request direction is to the partner system. This direction is specified for a – send request, i.e. the data is transferred to the partner. – request to view remote file attributes – request to view remote directories
	FROM	The request direction is to the local system (inbound). This direction is specified for a – receive request, i.e. the data are transferred to the local system – request to modify remote file attributes – request to delete remote files
	BOTH	File management request with two-way data transfer.
PROFILE	Name of the profile to be used for the transfer (empty in the FT logging record)	
TRANS-ID	Transfer ID number	
TRANSFER	Amount of data transferred	

Name	**Explanation**	
SEC-OPTS	Security options and encryption algorithms used. This line is only output if at least one of the options is used.	
	ENCR	Encryption of the request queue
	DICHK	Data integrity check of the request queue
	DENCR	Encryption of data content during the transfer
	DDICHK	Data integrity check of the file data to be transferred
	LAUTH	Authentication of the local system on a partner (authentication level 1)
	LAUTH2	Authentication of the local system on a partner (authentication 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.	
PTNR-ADDR	Address of the partner system, only output for inbound requests.	
FILENAME	Filename resp. preprocessing or postprocessing in local system. For security reasons, only the first 32 characters (or 42 characters in the case of FTEXECSV preprocessing operations) of a preprocessing or postprocessing command are taken over into the logging record. By arranging the call parameters accordingly or by inserting spaces, you can influence the command parameters that are not to appear in the logging record. FTEXECSV is the reaction to an ftexec command issued in a remote Windows or Unix system.	

Name	**Explanation**	
FNC-MODE	Encoding mode for file names and follow-up processing:	
	*TRANSPARENT	File names and follow-up processing are represented in a fixed binary code, independent of local character code settings (transparent mode).
	*CHAR, FNCCS=ccs	File names and follow-up processing are seen in their character presentation (character mode). ccs specifies the character set that was relevant for the creation of the FT request, e.g. iso88591 on Unix systems.
ADM-CMD	Only output for an ADM log record: Administration command without parameters	
ADMIN-ID	Only output for an ADM log record: Remains always empty in z/OS because only relevant on the remote administration server	
ROUTING	Only output for an ADM log record: Routing information on the openFT instance to be administered	
RC	Reason-Code. Indicates if a request was successfully executed, or if not, why it was rejected or terminated. If an FT request is rejected for “FTAC reasons” (e.g. 2169), the exact reason behind the termination can be found in the FTAC logging record of the system that rejected the request. Further information on the reason code can be obtained using the FTHELP xxxx command.	
REC-TYPE	Specifies if this is an FT or FTAC or ADM logging record.	
PCMD	Status of follow-up processing:	
	NONE	No follow-up processing defined.
	STARTED	Follow-up processing was started.
	NOT-STARTED	Follow-up could not be started.
PRIV	specifies whether the admission profile is privileged.	
WRITE	Write rules:	
	NEW	A new file is created. If a file with the same name already exists, the transfer will be aborted.
	EXT	An existing file is extended and stored as new.
	REPLACE	An existing file is extended.
TIME	Time when the logging record was written	

<table>
<tr><th>Name</th><th colspan="2">Explanation</th></tr>
<tr><td rowspan="2">FUNCTION</td><td colspan="2">Definition of FT function:</td></tr>
<tr><td colspan="2">– 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</td></tr>
<tr><td>STARTTIME</td><td colspan="2">Time request was started</td></tr>
<tr><td>STORETIME</td><td colspan="2">Time request was accepted (inbound)</td></tr>
<tr><td>REQUESTED</td><td colspan="2">Time request was accepted (outbound)</td></tr>
<tr><td>CCS-NAME</td><td colspan="2">Name of the character set, used for code conversion as necessary.</td></tr>
<tr><td rowspan="2">CHG-DATE</td><td colspan="2">Specifies whether the change date of the send file is taken over for the receive file.</td></tr>
<tr><td>SAME</td><td>The change date of the send file is take over.</td></tr>
<tr><td>INITSN</td><td colspan="2">TSN from which the request came, entered only in the case of outbound requests.</td></tr>
<tr><td>GLOB-ID</td><td colspan="2">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.</td></tr>
</table>

Examples

1. The FT administrator wants to display all logging records that were created for the user ID *Smith* and logged between 01.09.2016 and 31.12.2016. If you are the owner of the User ID *Smith*, you can omit the parameter OWNER-IDENTIFICATION=.

   ```
   FTSHWLOG SELECT=*PARAMETERS(OWNER-IDENTIFICATION=SMITH,          -
                    CREATION-TIME=*INTERVAL(FROM=2016-09-01(00:00), -
                    TO=2016-12-31(23:59))),NUMBER=*ALL
   ```

   You want to see the first record of the output in detail.

   ```
   FTSHWLOG (OWN=SMITH,CRE-TIME=*INTERVAL(FROM=2016-09-01(00:00),  -
             TO=2016-12-31(00:00))),INF=*ALL
   ```

2. An (FT or FTAC) administrator wants to view all log records. He/She wants all the information to be output in the most compact possible form because he/she wants to back up the log records before deleting them. To do this, he/she combines the specifications for "comprehensive output" and "output in CSV format". This is achieved using the following command:

   ```
   FTSHWLOG SELECT=*ALL,NUMBER=*ALL,INF=*ALL,OUTPUT=*STDOUT(*CSV)
   ```

   This command may take a few minutes to output comprehensive information.

3. The FT or FTAC administrator wishes to display the names of the current log file and and current offline log files:

   ```
   FTSHWLOG INF=*LOGGING-FILES
    'OPFTUID.STD.SYSLOG.L160802.L093109'
    logoff'OPFTUID.STD.SYSLOG.L160723.L061619'
   ```

4.33 FTSHWMON Show monitoring data

Note on usage

User group: FT user and FT administrator

Description of the function

The FTSHWMON command allows you to output the monitoring values from openFT operation on the local system. To do this, monitoring must be activated (see FTMODOPT command) and openFT must be activated.

Format

FTSHWMON
NAME = ***STD** / ***ALL** /<list-poss(100): alphanum-name 1..12>
,POLLING =***NONE** / ***PAR**AMETERS(...)
***PAR**AMETERS(...)
INTERVAL=**1** /<integer 1..600>
,NUMBER=***UNLIMITED** / <integer 1..3600>
,INFORMATION=***VALUES**(...) / ***TYPE**
***VALUES**(...)
DATA=***FORMATTED** / ***RAW**
,OUTPUT= ***STDERR**(...) / ***STDOUT**(...)
***STDERR**(...) / ***STDOUT**(...)
LAYOUT = ***STD** / ***CSV**

Operands

NAME =
Specifies what monitoring values are to be output.

NAME = *STD
A predefined default set of monitoring values is output, see “Examples” on page 309.

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 304).

POLLING =
Specifies the interval at which the monitoring values are to be polled.

POLLING =*NONE
The monitoring values are only polled once.

POLLING =*PARAMETERS
In this structure you specify a time interval and a repetition factor for polling the monitoring values. If an error occurs during polling, further repeated output is canceled.

> **INTERVAL = 1**
> The time interval for polling the monitoring values is 1 second.
>
> **INTERVAL = <integer 1..600>**
> Time interval in seconds for polling the monitoring values.
>
> **NUMBER = *UNLIMITED**
> There is no limit to the number of times the monitoring values are polled. To cancel the command, you can use the key combination PA1 and RESET, for example.
>
> **NUMBER = <integer 1..3600>**
> Here you specify how often the monitoring values are to be polled.

INFORMATION =
Specifies whether the monitoring values themselves or the type of the monitoring values is to be output.

INFORMATION = *VALUES(...)
The measured value is output. You can specify whether the monitoring values are to be output in formatted form or as raw data.

> **DATA =*FORMATTED**
> The monitoring values are formatted for visual display, e.g. as throughput, maximum or average.
>
> **DATA =*RAW**
> Raw, unformatted data is output. Monitoring values for the duration of an action are not output.

INFORMATION = *TYPE
Outputs the type and, where applicable, the scaling factor of the monitoring value or the type of the metadata.

The scaling factor is only of significance for some monitoring values and in CSV format if *RAW is not specified. In this case, the output value must be divided by the scaling factor to get the real value. In the case of formatted data in tabular format, the scaling factor 100 specifies that the number is output to 2 decimal places.

The following output values are possible for *TYPE:

*BOOL	Boolean value
*PERCENT	Percentage
*INT	Integer number (corresponds to *INT(1))
*INT(100)	Integer value with a scaling factor of 100
*TIME	Timestamp
*STRING	Text output for the selection

OUTPUT =
Output medium.

OUTPUT = *STDERR(...)
The data is output to SYSTSPRT or SYSERR, if this DDNAME is defined.

OUTPUT = *STDOUT(...)
The data is output to SYSPRINT.

LAYOUT = *STD
Output is formatted in a form readable by the user.
If the monitoring configuration changes (filters), a new header and a new start time for monitoring is output in standard output format.

LAYOUT = *CSV
Data is output in Character Separated Values format. This is a quasi-tabular format that is in widespread use in the field of PCs and in which the individual fields are separated by semicolons ";" (see page 435).

If the monitoring configuration changes (filters), the new start time for monitoring is shown in a separate column in CSV format.

4.33.1 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

ℹ If monitoring is deactivated for all partners (PARTNER-SELECTION=*NONE with FTMODOPT ...,MONITORING), only the following values are provided:

Status: StCLim, StCAct, StRqLim, StRqAct, StOftr, StFtmr, StFtpr, StTrcr

All the other values are set to 0.

Name	Meaning	Output with	Output unit	
			FORMATTED	RAW
ThNetbTtl	Throughput in net bytes: Number of bytes transferred	*STD/ *ALL	Number of bytes per second	Bytes, accumulated
ThNetbSnd	Throughput in net bytes (send requests): Number of bytes transferred with send requests	*STD/ *ALL	Number of bytes per second	Bytes, accumulated
ThNetbRcv	Throughput in net bytes (receive requests): Number of bytes transferred with receive requests	*STD/ *ALL	Number of bytes per second	Bytes, accumulated
ThNetbTxt	Throughput in net bytes (text files): Number of bytes transferred when transferring text files	*ALL	Number of bytes per second	Bytes, accumulated

Name	Meaning	Output with	Output unit	
			FORMATTED	RAW
ThNetbBin	Throughput in net bytes (binary files): Number of bytes transferred when transferring binary files	*ALL	Number of bytes per second	Bytes, accumulated
ThDiskTtl	Throughput in disk bytes: Number of bytes read from files or written to files with transfer requests	*STD/ *ALL	Number of bytes per second	Bytes, accumulated
ThDiskSnd	Throughput in disk bytes (send requests): Number of bytes read from files with send requests	*STD/ *ALL	Number of bytes per second	Bytes, accumulated
ThDiskRcv	Throughput in disk bytes (receive requests): Number of bytes written to files with receive requests	*STD/ *ALL	Number of bytes per second	Bytes, accumulated
ThDiskTxt	Throughput in disk bytes (text files): Number of bytes read from text files or written to text files with transfer requests	*ALL	Number of bytes per second	Bytes, accumulated
ThDiskBin	Throughput in disk bytes (binary files): Number of bytes read from binary files or written to binary files with transfer requests	*ALL	Number of bytes per second	Bytes, accumulated
ThRqto	openFT requests: Number of openFT requests received	*STD/ *ALL	Number per second	Accumulated number
ThRqft	File transfer requests: Number of file transfer requests received	*ALL	Number per second	Accumulated number
ThRqfm	File management requests: Number of file management requests received	*ALL	Number per second	Accumulated number
ThSuct	Successful requests: Number of successfully completed openFT requests	*STD/ *ALL	Number per second	Accumulated number
ThAbrt	Aborted requests: Number of aborted openFT requests	*STD/ *ALL	Number per second	Accumulated number
ThIntr	Interrupted requests: Number of interrupted openFT requests	*STD/ *ALL	Number per second	Accumulated number
ThUsrf	Requests from non-authorized users: Number of openFT requests in which the user check was terminated with errors	*STD/ *ALL	Number per second	Accumulated number

Name	**Meaning**	**Output with**	**Output unit**	
			FORMATTED	**RAW**
ThFoll	Started follow-up processing operations: Number of follow-up processing operations started	*ALL	Number per second	Accumulated number
ThCosu	Connections established: Number of connections successfully established	*ALL	Number per second	Accumulated number
ThCofl	Failed connection attempts: Number of attempts to establish a connection that failed with errors	*STD/ *ALL	Number per second	Accumulated number
ThCobr	Disconnections: Number of disconnections as a result of connection errors	*STD/ *ALL	Number per second	Accumulated number
DuRqtlOut	Maximum outbound request duration: Maximum request duration of an outbound request	*ALL	Milliseconds [1]	-
DuRqtlInb	Maximum inbound request duration: Maximum request duration of an inbound request	*ALL	Milliseconds [1]	-
DuRqftOut	Maximum outbound transfer request duration: Maximum duration of an outbound file transfer request	*ALL	Milliseconds [1]	-
DuRqftInb	Maximum inbound transfer request duration: Maximum duration of an inbound file transfer request	*ALL	Milliseconds [1]	-
DuRqfmOut	Maximum outbound file management request duration: Maximum duration of an outbound file management request	*ALL	Milliseconds [1]	-
DuRqfmInb	Maximum inbound file management request duration: Maximum duration of an inbound file management request	*ALL	Milliseconds [1]	-
DuRqesOut	Maximum outbound request waiting time: Maximum waiting time before an outbound request is processed (for requests without a specific start time)	*ALL	Milliseconds [1]	-

Name	Meaning	Output with	Output unit	
			FORMATTED	RAW
DuDnscOut	Maximum duration of an outbound DNS request Maximum time an outbound openFT request was waiting for partner checking	*ALL	Milliseconds [1]	-
DuDnscInb	Maximum duration of an inbound DNS request Maximum time an inbound openFT request was waiting for partner checking	*ALL	Milliseconds [1]	-
DuConnOut	Maximum duration of establishment of a connection: Maximum time between requesting a connection and receiving confirmation of a connection for an outbound openFT request	*ALL	Milliseconds [1]	-
DuOpenOut	Maximum file open time (outbound): Maximum time an outbound openFT request required to open the local file	*ALL	Milliseconds [1]	-
DuOpenInb	Maximum file open time (inbound): Maximum time an inbound openFT request required to open the local file	*ALL	Milliseconds [1]	-
DuClosOut	Maximum file close time (outbound): Maximum time an outbound openFT request required to close the local file	*ALL	Milliseconds [1]	-
DuClosInb	Maximum file close time (inbound): Maximum time an inbound openFT request required to close the local file	*ALL	Milliseconds [1]	-
DuUsrcOut	Maximum user check time (outbound): Maximum time an outbound openFT request required to check the user ID and transfer admission	*ALL	Milliseconds [1]	-
DuUsrcInb	Maximum user check time (inbound): Maximum time an inbound openFT request required to check the user ID and transfer admission	*ALL	Milliseconds [1]	-
StRqas	Number of synchronous requests in the ACTIVE state	*STD/ *ALL	Average [2]	Current number
StRqaa	Number of asynchronous requests in the ACTIVE state	*STD/ *ALL	Average value [2]	Current number
StRqwt	Number of requests in the WAIT state	*STD/ *ALL	Average value [2]	Current number

Name	Meaning	Output with	Output unit	
			FORMATTED	RAW
StRqhd	Number of requests in the HOLD state	*STD/ *ALL	Average value [2]	Current number
StRqsp	Number of requests in the SUSPEND state	*STD/ *ALL	Average value [2]	Current number
StRqlk	Number of requests in the LOCKED state	*STD/ *ALL	Average value [2]	Current number
StRqfi	Number of requests in the FINISHED state	*ALL	Average value [2]	Current number
StCLim	Maximum number of connections: Upper limit for the number of connections established for asynchronous requests.	*STD/ *ALL	Value currently set	
StCAct	Number of occupied connections for asynchronous requests	*STD/ *ALL	Share of StCLim in % [3]	Current number
StRqLim	Maximum number of requests: Maximum number of asynchronous requests in request management	*STD/ *ALL	Value currently set	
StRqAct	Entries occupied in request management	*STD/ *ALL	Share of StRqLim in % [3]	Current number
StOftr	openFT protocol activated/deactivated	*STD/ *ALL	ON (activated) OFF (deactivated)	
StFtmr	FTAM protocol activated/deactivated	*STD/ *ALL	ON (activated) OFF (deactivated)	
StFtpr	FTP protocol activated/deactivated	*STD/ *ALL	ON (activated) OFF (deactivated)	
StTrcr	Trace activated/deactivated	*ALL	ON (activated) OFF (deactivated)	

[1] Maximum value during the last monitoring interval (= time elapsed since the last time the monitoring values were queried or since the start of monitoring). The minimum time interval output is 1 millisecond if a relevant measurement has been completed during the interval since the last query. A value of 0 specifies that no measurement has been made in this interval.

[2] Average value during the monitoring interval (= time elapsed since the last time the monitoring values were queried or since the start of monitoring). The format is n.mm, where n is an integer and mm are to be interpreted as decimal places.

[3] If the reference value is reduced in live operation, it is possible for the value output to lie above 100 (%) temporarily.

4.33.2 Examples

1. Monitoring values are to be output in default output format.

```
FTSHWMON
openFT(STD)   Monitoring (formatted)
MonOn=2016-12-17 15:36:12 PartnerSel=OPENFT  RequestSel=ONLY-ASYNC,ONLY-LOCAL
2016-12-17 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 304 for the meanings.

2. Only the data types are to be output in default output format.

```
FTSHWMON INFORMATION=*TYPE
openFT(STD)   Monitoring (formatted)
MonOn=2016-12-17 15:36:12 PartnerSel=OPENFT  RequestSel=ONLY-ASYNC,ONLY-LOCAL
2016-12-17 15:40:01

 Name      Value
 ----------------
 ThNetbTtl INT
 ThNetbSnd INT
 ThNetbRcv INTs
 ThDiskTtl INT
 ThDiskSnd INT
 ThDiskRcv INT
 ThRqto    INT
 ThSuct    INT
 ThAbrt    INT
 ThIntr    INT
 ThUsrf    INT
 ThCofl    INT
 ThCobr    INT
 StRqas    INT(100)
 StRqaa    INT(100)
 StRqwt    INT(100)
 StRqhd    INT(100)
 StRqsp    INT(100)
 StRqlk    INT(100)
 StCLim    INT
 StCAct    PERCENT
 StRqLim   INT
 StRqAct   PERCENT
 StOftr    BOOL
 StFtmr    BOOL
 StFtpr    BOOL
```

Explanation

The types in the Value column have the following significance:

INT Integer number (corresponds to INT(1))

INT(100) Numeric value with a scaling value of 100 in the format n.mm, where n is an integer and mm are decimal places.

PERCENT Percentage

BOOL Boolean value, ON / OFF

3. The monitoring value "throughput in netbytes" (ThNetbTtl) is to be displayed. The display is to be updated every 60 seconds and repeated three times (polling).

   ```
   FTSHWMON
   NAME=ThNetbTtl,POLLING=*PAR(INTERVAL=60,NUMBER=3)
   ```

   ```
   openFT(STD)   Monitoring (formatted)
   MonOn=2016-12-16 10:44:09 PartnerSel=OPENFT,FTP  RequestSel=ONLY-ASYNC,ONLY-
   LOCAL
   2016-12-16 12:45:33
    Name      Values
    ----------------
    ThNetbTtl 780107

   2016-12-16 12:46:33
    ThNetbTtl 993051

   2016-12-16 12:47:33
    ThNetbTtl 1049832
   ```

   The repetitions are separated by intermediate header containing the current polling time.

4.34 FTSHWNET
Display the network environment

Note on usage

User group: FT administrator

This command has to be entered under TSO.

Functional description

You use this command to output information about the network environment of the current openFT instance.

Format

FTSHWNET

Without operands

Example

```
ftshwnet
openFT: VERSION    = 12.1A00
openFT: IP-ADDR    = 111.22.123.34
openFT: PORT-NR    = 1100
openFT: TCP-NAME   = TCPIP
openFT: INSTANCE   = STD
openFT: VTAM-FTID  = PBFT2
openFT: CMD-TRANS  = TCP
openFT: MSG-CRYPT  = N
openFT: SVC IN USE = 211
openFT: SS  IN USE = OPFT
```

4.35 FTSHWOPT Display operating parameters

Note on usage

User group: FT user and FT administrator

Functional description

The command FTSHWOPT can be used at any time to obtain the information listed below on the operating parameters of your FT system:

- Information on whether or not openFT has been started
- Instance identification
- Maximum values for operation (maximum number of file transfer requests in the request file, maximum lifetime of requests, maximum number of processes and transport connections, maximum size of a transport unit)
- Security settings (FTAC security level of the partner systems, extended sender verification)
- Logging settings (scope, intervals for automatic deletion)
- Trace settings
- Settings for traps (console traps, ADM traps)
- Settings for the monitoring functions

Format

FTSHWOPT
OUTPUT = ***STDERR**(...) / ***STDOUT**(...)
***STDERR**(...) / ***STDOUT**(...)
LAYOUT = ***STD** / ***CSV** / ***BS2-PROC** / ***ZOS-PROC**

Operands

OUTPUT =
Output medium.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or SYSERR, if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

LAYOUT = *STD
Output is put into a user-friendly form for reading.

LAYOUT = *CSV
Output takes place in **C**haracter **S**eparated **V**alues format. This is a special tabular format, widely used in the PC world, where the individual fields are separated by semicolons “;“ (see page 439).

LAYOUT = *BS2-PROC
The operating parameters are output as a command sequence. This can be called as an SDF procedure at BS2000 systems in order to recreate the identical operating parameters.

LAYOUT = *ZOS-PROC
The operating parameters are output as a command sequence. This can be called as a Clist procedure at z/OS systems in order to recreate the identical operating parameters.

4.35.1 Description of the output of FTSHWOPT

Example

Default of the FTSHWOPT command, i.e. the operating parameters have not been modified since installation.

```
STARTED PROC-LIM CONN-LIM ADM-CLIM RQ-LIM MAX-RQ-LIFE TU-SIZE  CCS-NAME
  YES       2       16        8     2000      30       65535    IBM1047
PTN-CHK DYN-PART SEC-LEV  FTAC-LOG FT-LOG ADM-LOG
  STD       ON   B-P-ATTR    ALL    ALL    ALL
OPENFT-APPL      FTAM-APPL         FTP-PORT           ADM-PORT
1100             *NONE             21                 11000
ACTIVE           NAVAIL            ACTIVE             ACTIVE
RSA-PROP RSA-MIN AES-MIN ENC-MAND
2048        0     NONE     NO
HOST-NAME        IDENTIFICATION / LOCAL SYSTEM NAME
MCHZPDT2         FJMPBFT2 / $FJAM,FJMPBFT2

DEL-LOG  ON  AT      RETPD    RECOVERY   ADM-TRAP-SERVER
 OFF   DAILY 00:00    14     IN+OUT       *NONE

TRAP: SS-STATE FT-STATE PART-STATE PART-UNREA RQ-STATE TRANS-SUCC TRANS-FAIL
CONS    OFF      OFF       OFF        OFF       OFF       OFF        OFF
ADM              OFF       OFF        OFF       OFF       OFF        OFF

FUNCT:  SWITCH PARTNER-SELECTION    REQUEST-SELECTION    OPTIONS
MONITOR  OFF   ALL                  ALL
TRACE    OFF   ALL                  ALL                  NONE
```

Meaning of the output fields

STARTED
Specifies whether openFT is activated (via FTSTART or automatically) or not.

PROC-LIM
Maximum number of tasks that can be reserved simultaneously for the execution of FT requests. The value is defined by the PROCESS-LIMIT operand in the FTMODOPT command.
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. The value of CONN-LIMIT is defined by the CONNECTION-LIMIT operand in the FTMODOPT command.

Default setting following installation: 16

ADM-CLIM
Maximum number of asynchronous administration requests including ADM traps that can be processed simultaneously. The value of ADM-CLIM is specified with the operand ADM-CONNECTION-LIM in the command FTMODOPT.
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. The value can be modified using the REQUEST-LIMIT operand in the FTMODOPT command.
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. The value is defined in the MAX-REQUEST-LIFETIME operand of the FTMODOPT command.
Default setting following installation: 30

TU-SIZE
Maximum size of a transport unit in bytes. The value is defined with the TRANSPORT-UNIT-SIZE operand in the FTMODOPT command. The load placed on the transport system by openFT can be controlled using this operand.
Default setting following installation: 65535

CCS-NAME
Name of the character set, which is used as standard character set for FT requests. The standard character set can be created with the CODED-CHARACTER-SET operand of the FTMODOPT command.
Default setting following installation: IBM1047

PTN-CHK
Defines whether or not enhanced sender checking is activated. The value is defined with the PARTNER-CHECK operand in the FTMODOPT command.
Default setting following installation: STD

DYN-PART
specifies whether dynamic partners are permitted (*ON) or not (*OFF). The value is defined with the DYNAMIC-PARTNERS operand in the FTMODOPT command.
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 using the SECURITY-LEVEL operand in the FTMODOPT command. All partners in the partner list for which the value STD is specified in the output of the FTSHWPTN command for SECLEV refer to this value.
This value is irrelevant for free dynamic partners (i.e. partner not entered in the partner list).
Default setting following installation: B-P-ATTR

FTAC-LOG
Scope for FTAC logging (ALL, MODIFY, REJECTED).
The scope of FTAC logging is specified in the LOGGING operand of the FTMODOPT command.
Default setting following installation: ALL

FT-LOG
Scope for FT logging (ALL, FAIL, NONE).
The scope of FT logging is specified in the LOGGING operand of the FTMODOPT command.
Default setting following installation: ALL

ADM-LOG
Scope of ADM logging (ALL, FAIL, MODIFY, NONE).
The scope of ADM logging is specified in the LOGGING operand of the FTMODOPT command.
Default setting following installation: ALL

OPENFT-APPL
Port number used by the local openFT. *STD means that the default port number 1100 is used. The value is specified with the OPENFT-APPLICATION operand in the command FTMODOPT.
The second line specifies whether the asynchronous inbound server is activated for openFT (ACTIVE), deactivated (DISABLED) or unavailable (INACT). The ACTIVE-APPLICATIONS operand in the command FTMODOPT is used for activation and deactivation.
Default setting following installation: *STD

FTAM-APPL
Not relevant on z/OS systems; is always supplied with *NONE.
Default setting following installation: *NONE

FTP-PORT
Port number used by the local FTP server. The value is specified with the FTP-PORT operand in the command FTMODOPT.
The second line specifies whether the asynchronous inbound server is activated for FTP (ACTIVE/DISABLED) or is unavailable or not installed (INACT/NAVAIL). The ACTIVE-APPLICATIONS operand in the command FTMODOPT is used for activation and deactivation.
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 value is specified with the ADM-PORT operand in the command FTMODOPT.
The second line specifies whether the asynchronous inbound server is activated for remote administration requests (ACTIVE), deactivated (DISABLED) or unavailable (INACT). The ACTIVE-APPLICATIONS operand in the command FTMODOPT is used for activation and deactivation.
Default setting following installation: 11000

RSA-PROP
Specifies the length of the RSA key used for encryption. This key is only used for the encryption of the AES key which the two partners have agreed upon (or the DES key). openFT uses the AES key to encrypt the request description data and possibly the file contents. The value is specified with the RSA-PROPOSED operand in the command FTMODOPT.
Default setting following installation: 2048

RSA-MIN
Specifies the minimum RSA key length (in bits) that is accepted by the communication partner for the transfer of the AES key of the session. The value is specified with the RSA-MINIMUM operand in the command FTMODOPT.
Default setting following installation: 0

AES-MIN
Specifies the minimum AES key length (in bits) that is required from a communication partner. The value is specified with the AES-MINIMUM operand in the command FTMODOPT.
Default setting following installation: NONE

ENC-MAND
Specifies whether user data encryption is mandatory for openFT requests.
The value can be modified with the ENCRYPTION-MANDATORY operand in the FTMODOPT command.
Default setting following installation: NO

HOST-NAME
Name of the host that is automatically taken over if you have specified a host during the FJGEN initialization run.
Default setting following installation: *NONE (if you specified no host name in the FJGEN initialization run)

IDENTIFICATION / LOCAL SYSTEM NAME
Instance identifier of the openFT instance currently set and the name of the local system. The instance identifier is defined with the IDENTIFICATION operand of the FTMODOPT command and is used to identify the instance in the partner systems.
Default setting following installation: The value is formed from the value for FT-ID which is transferred with FJGEN: FJM<ftid> / $FJAM,FJM<ftid>

DEL-LOG
Specifies whether automatic deletion of log records is activated.
The values can be modified using the DELETE-LOGGING operand in the FTMODOPT command.
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

RECOVERY
Specifies the global activation or deactivation of the recovery of outbound and inbound requests. The value can be modified with the RECOVERY-OUTBOUND and RECOVERY-INBOUND operands in the command FTMODOPT.
Default setting following installation: IN+OUT

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.
The value is specified with the ADM-TRAPS=(DESTINATION=...) operand in the command FTMODOPT.
Default setting following installation: *NONE

TRAP
This section with the rows CONS and ADM specifies the trap settings. The columns identify the events for which traps may be generated.

– SS-STATE: Subsystem state change (not for ADM traps)
– FT-STATE: State change of the openFT control process
– PART-STATE: Partner system state change

- PART-UNREA: Partner not reachable
- RQ-STATE: Request management state change
- TRANS-SUCC: Successfully completed requests
- TRANS-FAIL: Failed requests

The possible values are ON or OFF.
Default setting following installation: OFF (for all columns)

The following rows specify the settings for the various trap types:

CONS
Settings for console traps FTR03XXX. This is specified with the CONSOLE-TRAPS operand in the command FTMODOPT.

ADM
Setting for ADM traps to be output to the ADM trap server. This is specified with the ADM-TRAPS=(SELECTION=...) operand in the command FTMODOPT.

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. This is specified with the MONITORING operand in the command FTMODOPT.
Default setting following installation: OFF

TRACE
Setting for trace function. This is specified with the TRACE operand in the command FTMODOPT.
Default setting following installation: OFF

4.36 FTSHWPRF Display admission profile

Note on usage

User group: FTAC user and FTAC administrator

Prerequisite for using this command is the use of openFT-AC.

Functional description

With the command FTSHWPRF, FTAC users can obtain information about their admission profiles. The FTAC administrator can obtain information about all the admission profiles in his/her system.

Either the contents of the selected admission profile or only its name can be output. It is not possible to use FTSHWPRF to access passwords or transfer admissions defined in the profile! If a transfer admission is forgotten, a new one must be specified using FTMODPRF.

Format

```
FTSHWPRF

 NAME = *ALL / <alphanum-name 1..8> / *STD
,SELECT-PARAMETER = *OWN / *PARAMETERS(...)
   *PARAMETERS(...)
     | TRANSFER-ADMISSION = *ALL / *NOT-SPECIFIED / <alphanum-name 8..32> /
     |                              <c-string 8..32 with-low> / <x-string 15..64>
     |,OWNER-IDENTIFICATION = *OWN / *ALL / <name 1..8>
,INFORMATION = *ONLY-NAMES / *ALL
,OUTPUT = *STDERR(...) / *STDOUT(...)
   *STDERR(...) / *STDOUT(...)
     | LAYOUT = *STD / *CSV
```

Operands

NAME =
Name of the admission profile you wish to view.

NAME = *ALL
Views all admission profiles.

NAME = <alphanum-name 1..8>
Views the admission profile with the specified name.

NAME = *STD
Displays the standard 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
The transfer admission is not used as a selection criterion.

TRANSFER-ADMISSION = *NOT-SPECIFIED
Only admission profiles for which no transfer admission has been specified are displayed.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
Views the admission profile which can be addressed with this transfer admission.

OWNER-IDENTIFICATION =
Specifies, whose admission profiles you wish to view.

OWNER-IDENTIFICATION = *OWN
Views only your own admission profile.

OWNER-IDENTIFICATION = *ALL
The FTAC administrator can view all admission profiles, regardless of who the owner is.

OWNER-IDENTIFICATION = <name 1..8>
The FTAC user can only access his/her own admission profiles; the output corresponds to *OWN. The FTAC administrator can view the admission profiles of any FTAC user with this parameter.

INFORMATION =
Scope of information desired.

INFORMATION = *ONLY-NAMES
FTAC only outputs the name of the admission profile and indicates whether it is privileged or locked. An “*“ is output for privileged profiles and a “!“ for locked profiles.

INFORMATION = *ALL
FTAC outputs the contents of the admission profile, excluding any passwords and the transfer admission.
In the case of a blocked admission profile (marked with an "!" when output with INFORMATION=*ONLY-NAMES), the following values can appear in TRANS-ADM:

TRANS-ADM=	**Meaning**
(NOT-SPECIFIED)	No TRANSFER-ADMISSION specified in the admission profile.
(DUPLICATED)	The admission profile was blocked because the TRANSFER-ADMISSION was "detected" by another user and the profile was to be blocked in that case (USAGE=*PRIVATE is specified in the command FTCREPRF or FTMODPRF). "Detected" means that another user ID tried to assign the same TRANSFER-ADMISSION over again.
(LOCKED(by_user))	The admission profile was specifically blocked by the user (VALID=*NO was specified in the command FTCREPRF or FTMODPRF).
(LOCKED(by_adm))	The admission profile was specifically blocked by the FTAC administrator (VALID=*NO was specified in the command FTCREPRF or FTMODPRF).
(EXPIRED)	The validity of TRANSFER-ADMISSION has expired (EXPIRATION-DATE was specified in command FTCREPRF or FTMODPRF).

OUTPUT =
Output medium for the information.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or to SYSERR if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

LAYOUT = *STD
Output is formatted using a standard layout that can be easily read by the user.

LAYOUT = *CSV
Output is supplied in CSV (**C**haracter **S**eparated **V**alues) format. This is a widely used tabular format, especially in the PC environment, in which individual fields are separated by a delimiter, which is usually a semicolon “;“ (see page 444).

4.36.1 Description of the output of FTSHWPRF

Example 1

The FTAC administrator wishes to view the admission profile UMSAWARE of the user STEVEN with the command FTSHWPRF to determine if the profile might endanger data protection:

```
FTSHWPRF␣NAME=UMSAWARE, -
      SELECT-PARAMETER=(OWNER-IDENTIFICATION=STEVEN),INFORMATION=*ALL
```

Short form:

```
FTSHWPRF␣UMSAWARE,(,STEVEN),*ALL
```

The output takes the following form:

```
UMSAWARE
 EXP-DATE    = 20171031
 IGN-MAX-LEV = (IBR)
 FILE        = PROFIT
 USER-ADM    = (STEVEN,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  = 2016-10-11 13:38:11
```

The first line shows the name of the admission profile. EXP-DATE shows the expiration date of the admission profile. The next two lines show the settings which Steven made in the command FTCREPRF using the parameter IGNORE-MAX-LEVELS=(INBOUND-RECEIVE=*YES) and FILE-NAME= PROFIT. The values for USER-ADMISSION and PROCESSING-ADMISSION have not been set by Steven, but rather the default values have been used. The output SUCC-PROC=*NONE and FAIL-PROC=*NONE means that no follow-up processing is permitted. The output DATA-ENC=YES shows that Steven is especially careful, because this means that requests are only accepted if the user data is encrypted. Steven set this by using DATA-ENCRYPTION=*YES in the FTCREPRF command. The timestamp of the most recent change is shown under LAST-MODIF.

The timestamp is also updated if you do not change the properties of the profile, i.e. if you enter FTMODPRF only with the parameter NAME, but no other parameters.

i 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.

Example 2

The FTAC administrator examines the admission profile TESTPROF using the FTSHWPRF command to determine whether file processing is possible with this profile. The command is as follows:

```
FTSHWPRF˽NAME=TESTPROF, -
         SELECT-PARAMETER=(OWNER-IDENTIFICATION=STEVEN),INFORMATION=*ALL
```

Short form:

```
FTSHWPRF˽TESTPROF,(,STEVEN),INF=*ALL
```

The output has the following form:

```
TESTPROF
 INITIATOR   = REMOTE
 USER-ADM    = (STEVEN,*FIRST,OWN)
 PROC-ADM    = SAME
 FT-FUNCTION = (TRANSFER-FILE,FILE-PROCESSING)
 LAST-MODIF  = 2016-07-31 15:03:44
```

The first line of the output displays the name of the admission profile. The second line indicates that the profile can only be used for requests initiated in the remote system. Steven has specified the value *FIRST for ACCOUNT in USER-ADMISSION; this means that the first account number assigned to the home pubset of the specified user ID in the system is used for account assignment in the case of transfer requests. As a result, it is unaffected by any changes to the account number. However, Steven has not specified a value for PROCESSING-ADMISSION and the default value SAME is therefore used. This means that the values are taken over from USER-ADMISSION. The FT-FUNCATION line indicates that the examined profile supports both preprocessing and file transfer requests. The last row specifies when the profile was last modified. The timestamp is also updated if you do not change the properties of the profile, i.e. if you enter FTMODPRF only with the parameter NAME, but no other parameters.

Example 3

The FT administrator wishes to view the profile REMADMIN that has been set up for remote administration by a remote administrator.

```
FTSHWPRF˽NAME=REMADMIN,INFORMATION=*ALL
```

Output has the following form:

```
REMADMIN
 USER-ADM    = (BS2ADMIN,,YES)
 FT-FUNCTION = (REMOTE-ADMINISTRATION)
 LAST-MODIF  = 2016-06-25 15:31:29
```

The output REMOTE-ADMINISTRATION for FT-FUNCTION indicates that the profile is permitted to perform remote administration. This means that the profile can be used by remote administrators to administer the local openFT instance. These remote administrators must also be configured in the remote administration server.

4.37 FTSHWPTN Display partner systems

Note on usage

User group: FT user and FT administrator

Functional description

The FTSHWPTN command is used to obtain the following information on partner systems included in the partner list of the current openFT instance:

- the names of the remote systems in the partner list,
- the status of the 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. This security level applies only if FTAC functionality is used. The information can then also be obtained using the FTSHWRGE command.
- the number of not yet completed file transfer requests submitted in the local system,
- the number of file transfer requests submitted in the remote systems for the local system,
- the partner address,
- the type of sender checking,
- the recovery of outbound requests,
- in the case of output in CSV format: also the time of the last access and the authentication level.

i FTSHWPTN with the PARTNER=*ALL operand (default value) displays all **entered** dynamic partners. These can be recognized from the fact that they have no name. If you only want to output detailed information on one entered dynamic partner, you must specify the partner’s address in the PARTNER operand.
In the case of the FTSHWPTN command openFT does not check whether an address is valid. If, for example, you specify a random address of a free dynamic partner, this will be displayed with the default properties of a free dynamic partner.

Format

```
FTSHWPTN

 PARTNER = *ALL / <text 1..200 with-low>
,OUTPUT = *STDERR(...) / *STDOUT(...)
   *STDERR(...) / *STDOUT(...)
     |  LAYOUT = *STD / *CSV / *BS2-PROC / *ZOS-PROC
,STATE = *ALL / *ACTIVE / *DEACT / *INSTALLATION-ERROR / *NO-CONNECTION / *NOT-ACTIVE /
           *AUTOMATIC-DEACTIVATION / *INACTIVE-BY-AUTOMATIC-DEACT
,INFORMATION = *STD / *ALL
```

Operands

PARTNER =
Partner system or systems about which information is to be output.

PARTNER = *ALL
Information on all partner systems is output.

PARTNER = <text 1..200 with-low>
Name or address of the partner system or group of partner systems about which information is to be output.
If you enter a name then you have two options:
You can either enter a unique partner name (1 - 8 alphanumeric characters) or a group of partners identified by a 1 to 7-character specification followed by an asterisk (*).
For more information on partner addresses, see section “Specifying partner addresses” on page 76.

OUTPUT =
Output medium.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or SYSERR, if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

> **LAYOUT = *STD**
> Output is formatted using a standard layout that can be easily read by the user.
>
> **LAYOUT = *CSV**
> Output is supplied in CSV (**C**haracter **S**eparated **V**alues) format. This is a widely used tabular format, especially in the PC environment, in which individual fields are separated by a delimiter, which is usually a semicolon “;“ (see page 448).

LAYOUT = *BS2-PROC
Output is supplied in the form of MODIFY-FT-PARTNER commands, which precisely define the partners involved. This enables the partner entries to be saved for a later reconstruction, to use them for an openFT operation on BS2000.

LAYOUT = *ZOS-PROC
Output is supplied in the form of FTMODPTN commands, which precisely define the partners involved. This enables the partner entries to be saved for a later reconstruction, to use them for an openFT operation on z/OS (see example 2 in section “Description of the output of FTSHWPTN” on page 330).

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 330.

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.

4.37.1 Description of the output of FTSHWPTN

Example 1

Request information on all remote systems entered in the partner list:

Short output:

```
FTSHWPTN INF=*STD
NAME     STATE SECLEV   PRI  TRACE   LOC    REM P-CHK ADDRESS
         ACT    90      NORM FTOPT     0      0 FTOPT TEST011N
HOSTABS2 ACT   B-P-ATTR NORM FTOPT     0      0 FTOPT HOSTABS2
HOSTBBS2 ACT   STD      NORM FTOPT     0      0 FTOPT HOSTBBS2
PCUSER   ACT    40      LOW  FTOPT     0      0 FTOPT %IP123.23.99.120
PC1      ACT    40      LOW  FTOPT     0      0 FTOPT PC1
UNIX1    ACT    50      HIGH FTOPT     0      0 FTOPT UNIX1
UNIX2    ACT    50      HIGH FTOPT     0      0 FTOPT UNIX2:102
FTPUX1   ACT   STD      NORM FTOPT     0      0       ftp://%IP132.19.122.50
```

Long output:

```
FTSHWPTN INF=*ALL
NAME     STATE SECLEV   PRI  TRACE  LOC  REM P-CHK ADDRESS
         INBND REQU-P        RECOV                 ROUTING  IDENTIFICATION
         ACT   90       NORM FTOPT    0    0 FTOPT TEST011N
         ACT   STD           FTOPT                          TEST011N
HOSTABS2 ACT   B-P-ATTR NORM FTOPT    0    0 FTOPT HOSTABS2
         ACT   STD           ON                             HOSTABS2.FUJI.NET
HOSTBBS2 ACT   STD      NORM FTOPT    0    0 FTOPT HOSTBBS2
         ACT   STD           OFF                            HOSTBBS2.CLOUD.NET
PCUSER   ACT    40      LOW  FTOPT    0    0 FTOPT %IP123.23.99.120
         ACT   STD           FTOPT                          %IP123.23.99.120
PC1      ACT    40      LOW  FTOPT    0    0 FTOPT PC1
         ACT   STD           ON                             PC1.FUSI.NET
UNIX1    ACT    50      HIGH FTOPT    0    0 FTOPT UNIX1
         ACT   STD           FTOPT                          UNIX1.DREAM.NET
UNIX2    ACT    50      HIGH FTOPT    0    0 FTOPT UNIX2:102
         ACT   STD           OFF                            %.UNIX2.$FJAM
FTPUX1   ACT   STD      NORM FTOPT    0    0       ftp://%IP132.19.122.50
         ACT   STD           FTOPT
```

The information displayed is explained below:

NAME
Symbolic names of the remote systems entered in the partner list.
This field remains empty for dynamic partners (see the first line in the example).

STATE
Status of the partner system.

ACT
The partner system is active.

DEACT
The partner system is deactivated.

NOCON
The transport connection setup failed.

LUNK
The local system is unknown on the remote FT system.

RUNK
The partner system is unknown on the local transport system.

ADEAC
The partner system is active. It is deactivated if the connection cannot be established. This state is only displayed if STATE=*AUTOMATIC-DEACTIVATION has been specified; otherwise, these partner systems are maintained under the ACT status.

AINAC
The partner system was deactivated following several unsuccessful attempts to establish a connection. This status is only possible if STATE=*AUTOMATIC-DEACTIVATION has been specified.

LAUTH
The local system could not be authenticated in the partner system. A current, public key of the local openFT instance must be made available to the partner system.

RAUTH
The partner system could not be authenticated in the local system. A current, public key of the partner system must be imported to the SYSKEY library.

DIERR
A data integrity error was detected on the connection to the partner system. This can be due either to an error in the transport system, or to manipulation attempts along the transfer route. The connection was terminated but the affected request was not (if it is restartable).

NOKEY
The partner does not accept a connection without encryption, but no key is present in the local system. A new key must be created using FTCREKEY.

IDREJ
The partner or a go-between instance does not accept the instance ID sent from the local system. You must check to see if the local instance ID is consistent with the entry in the partner’s partner list.

SECLEV
Security level assigned to the remote system when it was entered in the partner list. These security levels apply only if the FTAC-BS2000 is also implemented. STD stands for the default security level set with the FTMODOPT command.

PRI
Priority of a partner with respect to the processing of requests. The possible values are NORM, LOW and HIGH.

TRACE
Trace setting. You may specify the values ON, OFF and FTOPT (if FTMODPTN is specified, TRACE=*BY-FT-OPTIONS).

LOC
Number of FT requests that have been submitted in the local system and that address the FT system specified with PARTNER.

REM
Number of FT requests that have been submitted in the remote FT system and addressed to the local FT system. The remote system is specified in PARTNER.

P-CHK
Type of sender checking for the current partner:

FTOPT
The global setting is valid.

T-A
The expanded sender checking is enabled for specific partners.

STD
The expanded sender checking is disabled for specific partners.

AUTH
With the aid of its public key in the SYSKEY library, the partner is subjected to an identity check (“authenticated”) by cryptographic means. The partner support the authentication level 2.

AUTH!
With the aid of its public key in the SYSKEY library, the partner is subjected to an identity check (“authenticated”) by cryptographic means. The partner support the authentication level 1.

NOKEY
No valid key is available from the partner system although authentication is required.

AUTHM
Authentication must be used.

ADDRESS
Partner address under which the remote system can be accessed. For more information on partner addresses, see section “Specifying partner addresses” on page 76.

IDENTIFICATION
Instance ID of the partner (also see the FTADDPTN command on page 98).

ROUTING
SESSION-ROUTING-INFO of the partner, where required (also see the FTADDPTN command on page 98).

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.

RECOV
Restart function (recovery) for outbound requests:

FTOPT
The global setting is valid.

ON
The restart function for outbound requests to this partner system is activated.

OFF
The restart function for outbound requests to this partner system is deactivated.

Example 2

All partner entries in the partner list are to be saved in a form that will facilitate importing the entries into a different partner list as required. To do this, the output from the FTSHWPTN command is converted to the correct format using LAYOUT=*ZOS-PROC and piped to a file with the name PARTZOS.CLIST.

```
 READY
FREE DDNAME(SYSPRINT)
 READY
ALLOC DSNAME(PARTZOS.CLIST) DDNAME(SYSPRINT) NEW KEEP DSORG(PS) RECFM(F,B)
      LRECL(80)
 READY
FTSHWPTN OUTPUT=*STDOUT(LAYOUT=*ZOS-PROC)
 READY
FREE DDNAME(SYSPRINT)
```

If the partner systems are to be entered in a partner list again, this can be done using the TSO command EXEC:

```
EXEC PARTZOS
```

This method also provides a simple way of importing partner entries from a z/OS partner list into a BS2000 partner list. To do this, LAYOUT=*BS2-PROC must be specified in FTSHWPTN and the file that is generated must be transferred to BS2000 and executed there. In the same way, a file created in BS2000 (as of openFT V9.0) using SHOW-FT-PARTNER can be used to enter partner systems in the z/OS partner list.

4.38 FTSHWRGE List partner systems

Note on usage

User group: FTAC user and FTAC administrator

Prerequisite for using this command is the use of openFT-AC.

Functional description

The command FTSHWRGE is used to list the partner systems with which you can communicate by file transfer. In addition to indicating the name of the partner system, the security level is output which the FT administrator assigned to this system in the partner list. To determine which basic functions you are permitted to use, you must use the command FTSHWADS to obtain information on your admission set (see page 270).

The FTAC administrator can use FTSHWRGE to list all partner systems with which his/her FT system can communicate using file transfer. Furthermore, he/she can find out for any user in his/her system which partner systems can be accessed by this user.

Format

```
FTSHWRGE

 USER-IDENTIFICATION = *OWN / <name 1..8>
,SELECT-PARAMETER = *ALL / *PARAMETERS(...)
    *PARAMETERS(...)
     |  PARTNER = *ALL / <text 1..200 with-low>
,OUTPUT = *STDERR(...) / *STDOUT(...)
    *STDERR(...) / *STDOUT(...)
     |  LAYOUT = *STD / *CSV
```

Operands

USER-IDENTIFICATION =
User ID for which you would like to have a list of accessible partner systems.

USER-IDENTIFICATION = *OWN
The FTAC user receives all the partner systems with which he/she can use at least one basic function.
The FTAC administrator receives all accessible partner systems.

USER-IDENTIFICATION = <name 1..8>
The FTAC user can only enter his/her own user ID here, the output corresponds to *OWN.
The FTAC administrator can enter any user ID for which he/she would like to view the accessible partner systems.

SELECT-PARAMETER =
Specifies selection criteria for the partner systems.

SELECT-PARAMETER = *ALL
Obtains information on all partner systems which can be reached.

SELECT-PARAMETER = *PARAMETERS(PARTNER = <text 1..200 with-low>)
Obtains information on this partner system. You can specify the name from the partner list or the address of the partner system. The following information is supplied:

- if you are permitted to communicate with this partner system.
- the security level assigned to this partner system.

For additional information to partner addresses, see section “Specifying partner addresses” on page 76.

OUTPUT =
Output medium for the partner system listing.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or to SYSERR if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

> **LAYOUT = *STD**
> Output is put into a user-friendly form for reading.
>
> **LAYOUT = *CSV**
> Output is in **C**haracter **S**eparated **V**alues format. This is a special tabular format, widely used in the PC world, where the individual fields are separated by a semicolon “;“ (see page 450).

Example

Steven Miller would like to find out about the security level of the computer BUYDACK. To do this, he uses the following command:

```
FTSHWRGE␣SELECT-PARAMETER=(PARTNER-NAME=BUYDACK)
```

Short form:

```
FTSHWRGE␣SEL=(BUYDACK)
```

He receives the following output:

```
SECLEV  PARTNER-NAME
  50    BUYDACK
```

The column SECLEV contains the security level of the partner system whose name appears in the PARTNER-NAME column.
If Steven had entered SELECT-PARAMETER=*ALL (or left out this parameter altogether), he would have received a similar but longer list of all accessible partner systems.

4.39 FTSTART Activate openFT

Note on usage

User group: FT administrator

Functional description

The FTSTART command is used to activate the specified openFT instance once the openFT load module has been loaded and started. If the value "A" for automatic activation was specified in the openFT start parameters (see the description of the FJGEN command in the manual "openFT (z/OS) - Installation and Operation") then it is not necessary to enter the FTSTART command.

The command is only executed if openFT is not already active.

If the request queue contains file transfer requests for which the corresponding (remote) FT systems have also been started, these requests are started directly after openFT starts – provided the resources are available and no other start time has been defined.

Adequate steps must also be taken to ensure that all file systems are available. Otherwise locally submitted requests that require unavailable file systems are terminated with an error message. If this happens, the user cannot be notified by a result list .

If the openFT instance is to run under a different host name, this host name must first be entered using FJGEN.

Format

FTSTART

Without operands

Correct execution of the FTSTART command is acknowledged with the following message:

```
FTR0500 OPENFT: openFT 12.1A00 starting. Protocols: openFT,FTP,ADM
```

4.40 FTSTOP Deactivate openFT

Note on usage

User group: FT administrator

Functional description

The FTSTOP is used to initiate deactivation of the specified openFT instance and stop openFT.

The command is only executed if the instance has been started.

Format

FTSTOP

Without operands

Correct execution of the FTSTOP command is acknowledged with the following message:

```
FTR0501 OPENFT: openFT terminated
```

Example

Activate the local openFT system and subsequently deactivate the FT system:

```
FTSTART
FTR0500 OPENFT: openFT 12.1A00 starting. Protocols: openFT, FTP, ADM
   .
   .
   .

FTSTOP
FTR0501 OPENFT: openFT terminated
```

4.41 FTTERM
Terminate openFT

Note on usage

User group: FT administrator

FTTERM can be entered in the TSO command mode only.

Functional description

You use the FTTERM command to terminate the openFT load module.

Format

Table 4:

FTTERM

Without operands

FTTERM is acknowledged with the following message:

```
FTR4131 OPENFT: TERMINATION INITIATED BY USER
```
or
```
FTR4121 OPENFT: TERMINATED
```

Notes

- If openFT is still active and FTTERM is entered, it is deactivated before being terminated.
- Reactivation of the FT system after an FTTERM command is achieved by entering the commands FJINIT and FTSTART.

4.42 FTUPDKEY Update public keys

Note on usage

User group: FT administrator

Functional description

Using the FTUPDKEY command, you can newly create the public key files of the key pair sets present in your openFT instance. This may become necessary if the existing public key files are unintentionally deleted. In addition, the command imports updated comments from SYSPKF.COMMENT to the public key files (see below).

The key pair consists of a private key, which is administered internally by openFT, and a public key.

Public keys are stored under the name:

<openft qualifier>.<inst>.SYSPKF.R<key reference>.L<key length>.

Here, the first two name parts are replaced by OPENFT QUALIFIER and the name of the instance.

The key reference is a numeric designator for the version of the key pair. Following installation, the key length is 2048 bits by default. The public key files are text files that are created in the character code of the respective operating system, i.e. EBCDIC.DF04-1 for BS2000, IBM1047 for z/OS, ISO8859-1 for Unix systems and CP1252 for Windows systems.

In a file <openft qualifier>.<inst>.SYSPKF.COMMENT, you can store comments that are written in the first lines of this file when an existing public key file is updated. Such comments might contain, for example, the communications partner and the telephone number of the FT administrator on duty. The lines in the SYSPKF.COMMENT file may be a maximum of 78 characters in length.

Public key files with invalid key reference are automatically deleted (for example, public keys, for which openFT no longer has an internal private key).

Format

FTUPDKEY

Without operands

4.43 NCANCEL Cancel file transfer requests

Note on usage

User group: FT user and FT administrator

Alias name: FTCANREQ

Functional description

The NCANCEL command can be used to cancel a file transfer request or to abort the file transfer. The FT system deletes from the request queue the file transfer request that corresponds to the specified selection criteria and, if necessary, aborts the associated file transfer.

The following features apply to this command:

- FT requests submitted either in the local or the remote system can be canceled.
- A single command can be used to cancel several FT requests simultaneously.
- The FT requests to be canceled can be selected using different selection criteria.
- The FT user can only cancel file transfer requests, whose "owner" he/she is.

  The owner of an FT request submitted in the local system is the user ID under which the request was issued.
  The owner of an FT request submitted in the remote system is the user ID that is accessed in the local system for the request.
- As FT administrator you can cancel requests from any user.
- As FT administrator you can also fully and unconditionally cancel a selected request and remove it from the request file. “Unconditional” means that, if necessary, the request can be cancelled without any negotiation with the corresponding partner system. In this way, you can clear the request file of requests which are no longer recognized in the partner system or for which there is no longer any connection to the partner system.

  WARNING!
  If not used carefully, this function can result in inconsistencies in the request files of the corresponding partner systems. Under certain circumstances these inconsistencies may cause baffling error messages (SYSTEM ERROR) and “dead requests” in the partner system request files. It should therefore only be used in exceptional circumstances and after a suitable period has elapsed.

After the FT request is canceled, openFT initiates a follow-up processing in the event of failure (FAILURE-PROCESSING) which was previously specified in the NCOPY command. The following points apply:

– If you cancel a request issued in the local system, local FAILURE-PROCESSING will be initiated in any case; FAILURE-PROCESSING will be initiated in the remote system only if the data transfer process had already begun.

– If you cancel a request issued in a partner system, FAILURE-PROCESSING will be initiated both in the local and the remote system, respectively.

Note

– The user who issued the file transfer request in the local system is informed that the request has been aborted, provided that the FAILURE-PROCESSING operand was used in the original transfer request to specify user-generated result information for the local system, or if the default result list is to be supplied.

– The user in the remote system is only informed that the file transfer request has been aborted if file transfer has already been started and if the FAILURE-PROCESSING operand has been used to request user-generated result information for the remote system.

– The file transfer requests aborted with NCANCEL remain in the request queue until both systems involved have informed each other of the abort action.

– Requests for which the file transfer proper has already been completed but where the decision to end the request has not yet been reached with the partner can no longer be canceled.

– If a request is canceled while preprocessing or postprocessing is running in z/OS, openFT starts a separate "Cancel-Job" to terminate the processing job. openFT constructs the cancel job with the TSOJOB job envelope from the openFT parameter library PARM. This job envelope is also required for follow-up processing (see the manual "openFT (z/OS) - Installation and Operation"). This Cancel-Job is assigned a "Z" as the last letter in the job name in order to give it a higher priority than the processing jobs that are currently running.

Format

NCANCEL / **FTCANREQ**

TRANSFER-ID = ***ALL** / <integer 1..2147483647> (**FORCE**-**CAN**CELLATION = ***NO** / ***YES**)
,**SEL**ECT = ***OWN** / ***PAR**AMETERS(...)
   ***PAR**AMETERS(...)
      OWNER-IDENTIFICATION = ***OWN** / ***ALL** / <name 1..8>
      ,**INIT**IATOR = (***LOC**AL, ***REMOTE**) / list-poss(2): ***LOC**AL / ***REMOTE**
      ,**PART**NER = ***ALL** / <text 1..200 with-low>
      ,**FILE-NAME** = ***ALL** / <filename 1..59> / <c-string 1..512 with-low>

You cannot issue an NCANCEL command without specifying any operands. In order to cancel or withdraw all your FT requests you may enter, for example:

```
NCANCEL *ALL
```

This is intended to prevent you unintentionally canceling all your FT requests by accidentally issuing an NCANCEL command without specifying any operands.

Operands

TRANSFER-ID =
Transfer ID of the FT request to be canceled.

TRANSFER-ID = *ALL
Deletes all FT requests if no further selection criteria are specified with SELECT. FT users can only delete FT requests of their own ID using this entry. FT administrators can delete all current FT requests that access the system.

TRANSFER-ID = <integer 1..2147483647>
Request identification which was communicated to the local system in the FT request confirmation. The associated FORCE-CANCELLATION parameter is available only to the FT administrator. It is used for an **unconditional** request cancellation.

TRANSFER-ID = <integer 1..2147483647>(FORCE-CANCELLATION = *NO)
NO is the default value. The request is removed from the request file following negotiation with the partner system.

TRANSFER-ID = <integer 1..2147483647>(FORCE-CANCELLATION = *YES)
The request is removed from the request file without negotiation with the partner system. This specification is only possible for an FT administrator who has previously attempted to cancel the request with NCANCEL <transfer-id> (FORCE-CAN=*NO).

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 = *ALL
Cancels FT requests under all user IDs. Only the administrator can use this entry.

OWNER-IDENTIFICATION = <name 1..8>
Specifies a particular user ID whose FT requests are to be canceled. As FT user you can only specify your own ID.

INITIATOR =
Initiator of the FT requests to be canceled.

INITIATOR = (*LOCAL,*REMOTE)
Cancels FT requests in the local system and in remote systems.

INITIATOR = *LOCAL
Cancels FT requests issued in the local system.

INITIATOR = *REMOTE
Cancels FT requests issued in remote systems.

PARTNER =
Cancels FT requests that were to be executed with a specific partner system.

PARTNER = *ALL
The name of the partner system is not used as a selection criterion to determine the FT requests to be canceled.

PARTNER = <text 1..200 with-low>
The FT requests that were to be executed with this partner are to be canceled.
The name must be specified in the same form in which it is output using NSTATUS.

FILE-NAME =
Cancels all FT requests in the local system that access this file or this library element whether as a send file or receive file. The file name or library member name must be specified exactly as it appears in the file transfer request.

FILE-NAME = *ALL
The file name is not used as a selection criterion to determine the FT requests to be canceled.

FILE-NAME = <filename 1..59> / <c-string 1..512 with-low>
Cancels FT requests in the local system that access this file.

If multiple selection criteria are specified in the NCANCEL command, then each one of these must be valid for the requests that are to be canceled. Otherwise the NCANCEL command is acknowledged with the following message:

```
FTR0504 OPENFT: No requests available for the selection criteria.
```

Examples

1. An FT user wants to cancel all FT requests which carry his/her own ID. In order to do this, it is sufficient to issue the command only with the operand *ALL:

   ```
   NCANCEL *ALL
   ```

   If only one job was existent, openFT acknowledges the request with the following message:

   ```
   FTR2072 Request 12334456 has been canceled
   ```

   If several jobs were existent, all requests are deleted with no prompt for confirmation and cancellation is acknowledged by messages:

   ```
   FTR2072 Request 12334558 has been canceled

   FTR2072 Request 12334739 has been canceled

   FTR2072 Request 12339336 has been canceled
   ```

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

   ```
   NCANCEL TRANSFER-ID=194578
   ```

   The recommended short form of this command is as follows:

   ```
   NCANCEL 194578
   ```

3. As an FT administrator, you want to cancel the request with the transfer ID 194578; you know that the user ID USER1 is the owner of this FT request. You issue the following command:

   ```
   NCANCEL TRANSFER-ID=194578,SELECT=(OWNER=USER1)
   ```

   A possible short form of this command is as follows:

   ```
   NCAN 194578,(USER1)
   ```

   openFT acknowledges the request with the following message:

   ```
   FTR2072 OPENFT: Request 194578 has been canceled
   ```

4. An FT user wishes to cancel all file transfer requests from remote system VAR001 that access his/her file DATA. This can be achieved with the following command:

   ```
   NCANCEL TRANSDER-ID=*ALL,SELECT=(INITIATOR=*REMOTE,PARTNER=VAR001,
   FILE-NAME=DATA)
   ```

   The recommended short form of this command is as follows:

   ```
   NCANCEL *ALL,SEL=(INIT=*REMOTE,PARTNER=VAR001,FILE=DATA)
   ```

5. As an FT administrator, you want to cancel the request with the transfer ID 655423; you do not know who is the owner of this FT request. You issue the following command:

   ```
   NCANCEL TRANSFER-ID=655423,SELECT=(OWNER=*ALL)
   ```

   A possible short form of this command is as follows:

   ```
   NCAN 655423,(*ALL)
   ```

6. As an FT administrator, you want to cancel **all** FT requests involving your FT system. You issue the following command:

   ```
   NCANCEL TRANSFER-ID=*ALL,SELECT=(OWNER=*ALL)
   ```

   A possible short form of this command is as follows:

   ```
   NCAN *ALL,(*ALL)
   ```

4.44 NCOPY Transfer file asynchronously

Note on usage

User group: FT user

Alias name: FTACOPY

Functional description

The NCOPY command serves to transfer sequential files (PS data sets and generation data sets of this type), "entry sequenced" VSAM files, individual members of PO or PDSE data sets (libraries) and entire PO or PDSE data sets. In addition, openFT can also access migrated files in z/OS and transfer them to the remote system.

The local system is regarded as the system in which the command is issued, or in this case, the z/OS computer. The partner system is designated as the remote system.

4.44.1 Introduction to the NCOPY command

If you wish to transfer a file, you must first indicate whether you wish to send (TO) the file or receive (FROM) it by using the operand TRANSFER-DIRECTION.

Following this the PARTNER operand is used to define the system with which the transfer is to take place.

The next step is to define the characteristics of the local system by using the LOCAL-PARAMETER operand. The structure specifications for the LOCAL-PARAMETER are to be entered in parentheses, i.e. LOCAL-PARAMETER=(...).

The REMOTE-PARAMETER operand contains details of the remote system. The structure specifications for the REMOTE-PARAMETER must also be entered in parentheses, i.e. REMOTE-PARAMETER=(...). In addition, the partner system type may also be specified before these parentheses; the possible entries are *BS2000, *MSP (for a partner system with z/OS) or *ANY (see page 365).

The remaining “optional” operands (see page 375) are used to define the other characteristics of the file transfer, such as compressed or encrypted transfer or the starting time for the transfer.

4.44.1.1 The shortest form of the command

The mandatory parameters for the NCOPY command are the entries for

- direction of transfer
- name of the remote system
- name of the file in the local system (mandatory if FTAC is not implemented)
- name of the file in the remote system
- remote TRANSFER-ADMISSION.

A file transfer can be effected using these three parameters alone, if:

- the send and receive files are not password-protected

An example can be found on page 382.

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 as of page 353.

4.44.1.2 How to find out if the file transfer request has been executed

The command NSTATUS can be used to establish the status of file transfer requests that are not yet complete. On completion of a transfer, the result is stored in a logging record.

It is also possible to use the NCOPY command to request that a result message be generated. There are three ways of generating such a message:

- allow the result message to be created by the system
- have a user-generated result message output as follow-up processing

A system-generated message can only be requested in the local system. This is achieved using the LISTING operand which enables you for example to order a result list in all cases (LISTING=*PARAMETER(CONDITION=ANY)), or to order a result list only when the file transfer is aborted due to an error (LISTING=*PARAMETER(CONDITION=ON-FAILURE-ONLY)). The result list can be output to SYSLST or to a file. By default, no result list is created.

If a result list is printed then the user whose user ID was specified in the local TRANSFER-ADMISSION is informed of the termination of the file transfer job by means of an asynchronous message (NOTIFY message in z/OS).

Follow-up processing can also be requested in the NCOPY command. There are four types of follow-up processing:

- follow-up processing in the local system if the file transfer has been successfully completed
- follow-up processing in the remote system if the file transfer has been successfully completed

- follow-up processing in the local system if the file transfer has been aborted because of an error
- follow-up processing in the remote system if the file transfer has been aborted because of an error.

Follow-up processing after a successful file transfer can be defined for both systems by the operand SUCCESS-PROCESSING, while that following a failed file transfer is defined by FAILURE-PROCESSING.

If follow-up processing is to take place under a different user ID from that specified by TRANSFER-ADMISSION, then that user ID can be specified using the PROCESSING-ADMISSION operand.

Example

In this example described in the previous section the CLIST procedure 'CAESAR.MISTAKE.CLIST' is to be executed in the local system under the ID CAESAR with the account number ACCT0003 and the password ABCDEFGH if the file transfer was not successful. In addition, a result list is to be printed under the ID FRED with account number ACCT0001 (without a password). If the file transfer has been successful, a message is to be sent to the user BERT in the remote system with account number ACCT0002 and password P1234567. In addition the file is to be assigned the name DATA and stored under the user ID BERT.

```
NCOPY
TRANSFER-DIRECTION=TO,
PARTNER=VAR001,
LOCAL-PARAMETER=(FILE=DATA,
    TRANSFER-ADMISSION=(USERIDENTIFICATION=ANTON,
        ACCOUNT=ACCT0001,PASSWORD=HUGO),
    PROCESSING-ADMISSION=(USER-IDENTIFICATION=CAESAR,
        ACCOUNT=ACCT0003,PASSWORD=ABCDEFGH),
    FAILURE-PROCESSING='EX IRRTUM',LISTING=*SYSLST),
REMOTE-PARAMETER=*MSP(FILE=DATEN,
    TRANSFER-ADMISSION=(USER-IDENTIFICATION=BERT,
        ACCOUNT=ACCT0002,PASSWORD=P1234567),
    SUCCESS-PROCESSING='SEND ''FILE TRANSFER O.K.'',USER(*)')
```

A possible short form of this command is as follows:

```
NCOPY TRANS=TO,PARTNER=VAR001,
LOC=(FILE=DATA,TRANS=(ANTON,ACCT0001.HUGO),
PROC=(CAESAR,ACCT0003,ABCDEFGH),FAIL='EX IRRTUM'),
REM=*MSP(FILE=DATEN,TRANS=(BERT,ACCT0002,P1234567),
SUCC='SEND ''FILE TRANSFER O.K.'',USER(*)')
```

The commands defining follow-up processing must be enclosed in single quotes. Any quotes specified within the command must be doubled (see SEND command in the example above).

If neither a result list nor follow-up processing has been requested for a request, you can use the logging function to determine whether the request has been carried out. A request may generate up to four logging records:

- an FT logging record in the initiator system on request termination. A precondition for this is that the request has been correctly accepted by openFT and FT logging is active.
- an FT logging record in the responder system on request termination. A precondition for this is that the actual file transfer operation has already been concluded and FT logging is active.
- an FTAC logging record in the initiator system on acceptance of the request. A precondition for this is that openFT-AC is used in the initiator.
- an FTAC logging record in the responder system on acceptance of the request. A precondition for this is that openFT-AC is used in the responder.

You view logging records with FTSHWLOG (see page 281).

4.44.2 Full form of the NCOPY command

Format

(part 1 of 5)

NCOPY / FTACOPY
TRANSFER-**DIR**ECTION = **TO-**PARTNER / **FROM-**PARTNER ,**PART**NER = <text 1..200 with-low>

(part 2 of 5)

```
,LOCAL-PARAMETER = *PARAMETERS(...)

  *PARAMETERS(...)
   |  FILE-NAME = *NOT-SPECIFIED / <filename 1..59> / <c-string 1..512 with-low>
   |  ,PASSWORD = *NONE / <alphanum-name 1..8>
   |  ,TRANSFER-ADMISSION = *SAME / <alphanum-name 8..32> / <x-string 15..64> /
   |                            <c-string 8..32 with-low> / *PARAMETERS(...)
   |    *PARAMETERS(...)
   |     |  USER-IDENTIFICATION = <name 1..8>
   |     | ,ACCOUNT = *NONE / <alphanum-name 1..40> / c-string 1..40>
   |  ,PASSWORD = *NONE / <alphanum-name 1..8>,PROCESSING-ADMISSION = *SAME / *NOT-
SPECIFIED / *PARAMETERS(...)
   |    *PARAMETERS(...)
   |     |  USER-IDENTIFICATION = <name 1..8>
   |     | ,ACCOUNT = *NONE / <alphanum-name 1..40> / c-string 1..40>
   |  ,PASSWORD = *NONE / <alphanum-name 1..8>,SUCCESS-PROCESSING = *NONE /<c-string
1..1000 with-low>
   |  ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
   |  ,LISTING = *STD / *NONE / *SYSLST / *LISTFILE / *PARAMETERS(...)
   |    *PARAMETERS(...)
   |     |  OUTPUT = *SYSLST / *LISTFILE
   |     | ,CONDITION = *ANY / *ON-FAILURE-ONLY
   |  ,CODED-CHARACTER-SET = *STD / <alphanum-name 1..8>
```

(part 3 of 5)

```
,REMOTE-PARAMETER = *BS2000(...) / *MSP(...) / *ANY(...)

  *BS2000(...)
        FILE-NAME = *NOT-SPECIFIED / <filename 1..54> / <c-string 1..512 with-low> /
                    *LIBRARY-ELEMENT(...)
          *LIBRARY-ELEMENT(...)
              LIBRARY = *NOT-SPECIFIED / <filename 1..54>
             ,ELEMENT = *NOT-SPECIFIED /
                        <filename 1..64 without-gen-vers>(...) / <composed-name 1..64 with-under>(...)
                 <filename>(...) / <composed-name>(...)
                      VERSION = *STD / <text 1..24>
             ,TYPE = *NOT-SPECIFIED / <name 1..8>
       ,PASSWORD = *SAME / *NONE / <c-string 1..4> / <x-string 1..8> /
                   <integer -2147483648..2147483647>
       ,TRANSFER-ADMISSION = <alphanum-name 8..32> / <x-string 15..64> /
                             <c-string 8..32 with-low> / *PARAMETERS(...)
          *PARAMETERS(...)
              USER-IDENTIFICATION = <name 1..8>
             ,ACCOUNT = *NONE / <alphanum-name 1..8>
             ,PASSWORD = *NONE / <c-string 1..8> / <c-string 9..32> / <x-string 1..16>
       ,PROCESSING-ADMISSION = *SAME / *NOT-SPECIFIED / *PARAMETERS(...)
          *PARAMETERS(...)
              USER-IDENTIFICATION = <alphanum-name 1..8>
             ,ACCOUNT = *NONE / <alphanum-name 1..8>
             ,PASSWORD = *NONE / <c-string 1..8> / <c-string 9..32> / <x-string 1..16>
       ,SUCCESS-PROCESSING = *NONE / <c-string 1..1000 with-low>
       ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
       ,CODED-CHARACTER-SET = *STD / <alphanum-name 1..8>
```

(part 4 of 5)

```
*MSP(...)
   FILE-NAME = *NOT-SPECIFIED / <filename 1..59> / <c-string 1..512 with-low>
  ,PASSWORD = *NONE / <alphanum-name 1..8>
  ,TRANSFER-ADMISSION = <alphanum-name 8..32> / <x-string 15..64> /  <c-string 8..32 with-low> /
                        *PARAMETERS(...)
     *PARAMETERS(...)
        USER-IDENTIFICATION = <name 1..8>
       ,ACCOUNT = *NONE / <alphanum-name 1..40> / <c-string 1..40>
       ,PASSWORD = *NONE / <alphanum-name 1..8>
  ,PROCESSING-ADMISSION = *SAME / *NOT-SPECIFIED / *PARAMETERS(...)
     *PARAMETERS(...)
        USER-IDENTIFICATION = <name 1..8>
       ,ACCOUNT = *NONE / <alphanum-name 1..40> / <c-string 1..40>
       ,PASSWORD = *NONE / <alphanum-name 1..8>
  ,SUCCESS-PROCESSING = *NONE /<c-string 1..1000 with-low>
  ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
  ,CODED-CHARACTER-SET = *STD / <alphanum-name 1..8>

*ANY(...)
   FILE-NAME = *NOT-SPECIFIED / <c-string 1..512 with-low> /
               *LIBRARY-ELEMENT(...)
     *LIBRARY-ELEMENT(...)
        LIBRARY = *NOT-SPECIFIED / <c-string 1..63 with-low>
       ,ELEMENT = *NOT-SPECIFIED / <c-string 1..64 with-low>(...)
          <c-string 1..64 with-low>(...)
             VERSION = *NONE / *STD / <c-string 1..24 with-low>
       ,TYPE = *NONE / *NOT-SPECIFIED / <c-string 1..8 with-low>
  ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128>
  ,TRANSFER-ADMISSION = *NONE / <alphanum-name 8..32> / <x-string 15..64> /
                        <c-string 8..32 with-low> / *PARAMETERS(...)
     *PARAMETERS(...)
        USER-IDENTIFICATION = <c-string 1..67 with-low>
       ,ACCOUNT = *NONE / <c-string 1..64 with-low>
       ,PASSWORD = *NONE / <c-string 1..64 with-low> /  <x-string 1..128 with-low>
  ,PROCESSING-ADMISSION = *SAME / *NONE / *PARAMETERS(...)
     *PARAMETERS(...)
        USER-IDENTIFICATION = <c-string 1..67 with-low>
       ,ACCOUNT = *NONE / <c-string 1..64 with-low>
       ,PASSWORD = *NONE / <c-string 1..64 with-low> / <x-string 1..128 with-low>
```

(part 5 of 5)

```
         | ,SUCCESS-PROCESSING = *NONE /<c-string 1..1000 with-low>
         | ,FAILURE-PROCESSING = *NONE / <c-string 1..1000 with-low>
         | ,CODED-CHARACTER-SET = *STD / <c-string 1..8 with-low>

,COMPRESS = *NONE / *BYTE-REPETITION / *ZIP

,WRITE-MODE = *REPLACE-FILE / *NEW-FILE / *EXTEND-FILE

,DATA-TYPE = *NOT-SPECIFIED / *CHARACTER (...) / *BINARY (...) / *USER

    *CHARACTER(...)
         |  TRANSPARENT = *NO / *YES

    *BINARY(...)
         |  TRANSPARENT = *NO / *YES

,PRIORITY = *NORMAL / *HIGH / *LOW

,START = *SOON / *EARLIEST(...)

    *EARLIEST(...)
         |  DATE = *TODAY / *TOMORROW / <date 8..10>
         | ,TIME = 00:00 / <time 1..8>

,CANCEL = *NO / *AT(...)

    *AT(...)
         |  DATE = *TODAY / *TOMORROW / <date 8..10>
         | ,TIME = 23:59 / <time 1..8>

,DATA-ENCRYPTION = *NO / *YES / *ONLY-DATA-INTEGRITY

,RECORD-SIZE = *NOT-SPECIFIED /<integer 1..32756>
         | ,RECORD-FORMAT = *STD / *FIXED / *VARIABLE / *UNDEFINED

,TABULATOR = *AUTO / *ON / *OFF

,TARGET-FILE-FORMAT = *SAME / *BLOCK-ORIENTED / *SEQUENTIAL(...)

    *SEQUENTIAL(…)
         | RECORD-FORMAT = *SAME / *UNDEFINED
```

Operands

TRANSFER-DIRECTION =

Direction of transfer.

TRANSFER-DIRECTION = TO-PARTNER

The local system is the send system. The files are dispatched to the partner system.

TRANSFER-DIRECTION = FROM-PARTNER

The local system is the receive system. The files are obtained from the partner system.

PARTNER = <text 1..200 with-low>
Name of the partner system as defined by the FT administrator in the partner list or the address of the partner system. For more information on address specifications, see section “Specifying partner addresses” on page 76.

Specifications for the local system (LOCAL-PARAMETER)

LOCAL-PARAMETER = *PARAMETERS(...)
Specifications for the local system.

FILE-NAME =
Name of the file in the local system (send file or receive file).

FILE-NAME = *NOT-SPECIFIED
The name of the file is known locally because it has already been completely defined in the FTAC admission profile addressed locally.

FILE-NAME = <filename 1..59> / <c-string 1..512 with-low>
When sending, the name of the file or preprocessing command, or, when receiving, the name of the postprocessing command. The specifications differ for with and without pre- and postprocessing.

*Specifications **without** pre- or postprocessing on FILE-NAME*

All types of filename can be either fully qualified or partially qualified:

- Fully qualified specification: the filename is enclosed in single quotes; the "first level qualifier" is the user ID or the alias under which the file is or will be cataloged.
- Partially qualified specification: the filename is not enclosed in single quotes; in this case, openFT adds the user ID for which the file transfer is being performed (TRANSFER-ADMISSION operand) as the "first level qualifier".
- If you use the c-string data type for a partially qualified filename, you must specify it in the form C’FILE.XYZ’.

If complete PO or PDSE data sets are to be transferred, the receive file only needs to be specified in the request with the corresponding type :O: or :E: openFT with a version < 10 is running on the partner system.

openFT permits the automatic generation of unique filenames as a simple way of preventing conflicts. You do this by entering %UNIQUE in the filename.

Specifications ***with*** *pre- or postprocessing on FILE-NAME*

- If you specify preprocessing commands on a send operation, the specified commands are first started as a TSO job. The data is output via the %TEMPFILE variable or via SYSPRINT to a temporary file ("preprocessing")
- If you specify postprocessing commands on a receive operation, openFT provides the first command with the transferred data via the %TEMPFILE variable or via SYSUT1 and waits until processing is concluded ("postprocessing").

For both pre- and postprocessing, a c-string must be specified on FILE-NAME. The first character must be a pipe symbol ‘|’, followed by the command string. If several commands are specified, they must be separated by a semicolon (‘;’).

Example

```
    FILE-NAME=C‘|Command1;Command2;Command3; ...‘
```

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

```
command1;WHEN SYSRC(< 12) command2;WHEN SYSRC(< 12) command3;...
```

The total maximum length of commands is restricted to the maximum file name length. For more information refer to the section “Preprocessing and postprocessing” on page 81. Also refer to the topic “Preprocessing” at the example starting on page 382.

If an error occurs during command execution, transfer is aborted with message FTR2206 or FTR2207.

If a transfer request with preprocessing and postprocessing is to be restartable, the characters ‘|&’ must be specified instead of ‘|’. For more details, also see section “Preprocessing and postprocessing” on page 81.

Example

```
    FILE-NAME = C‘|&command1;command2;command3; ...‘
```

To prove admission for preprocessing or postprocessing, the local TRANSFER-ADMISSION must either be explicitly supplied with USER-ID, ACCOUNT and PASSWORD or implicitly supplied with these specifications via a transfer admission belonging to an admission profile that contains them.

PASSWORD =
Password authorizing access to the file in the local system. If the file in the local system is password-protected (by means of the TSO command "PROTECT"), the password must be specified in this operand as:

- a write password for a receive file, or
- a read password for a send or receive file that is not protected by a write password but by a read password,

Newly created receive files are not given a password by this operand. PASSWORD is ignored in such cases.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <alphanum-name 1..8>
Password authorizing access to the file in the local system.

TRANSFER-ADMISSION =
Transfer admission of the user for the local system.

TRANSFER-ADMISSION = *SAME
The ID of the user entering the command is valid for the file transfer.

i To prove admission for preprocessing or postprocessing, the local TRANSFER-ADMISSION must either be explicitly supplied with USER-ID, ACCOUNT and PASSWORD or implicitly supplied with these specifications via a transfer admission belonging to an admission profile that contains them.
The same applies if migrated files are to be transferred to a remote system. If the admission is missing in the case of migrated files, message FTR2029 is issued.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
If the FTAC functionality is used the file name for the local system can be defined in an FT profile. The transfer admission defined in the FT profile must be specified here. From this transfer admission the access rights in the local system can be defined. These access rights are also defined in the FT profile. The alphanumeric entry is converted internally to lowercase characters.

The alphanumeric entries are converted internally to lowercase characters.

When entering the transfer admission in the form of a c-string you must remember that in z/OS a transfer admission defined as a c-string is stored using uppercase characters in the profile. To go with this you must also enter it in the NCOPY command, to ensure that conversion to lowercase is avoided. Here, you should bear in mind the notes regarding the distinction between uppercase and lowercase characters which can be found on page 40.

TRANSFER-ADMISSION = *PARAMETERS(...)
User ID, account number and password under which file transfer in the local system is to be performed. The operands in parentheses can also be used a positional operands without their keywords.

USER-IDENTIFICATION = <name 1..8>
User ID in the local system.

ACCOUNT =
Account number under which file transfer is performed in the local system.

ACCOUNT = *NONE
The default account number of the user ID is used.

ACCOUNT = <alphanum-name 1..40> / <c-string 1..40>
Account number of the user in the local system.

PASSWORD =
Password authorizing the user to access the local system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <alphanum-name 1..8>
Password that authorizes the user to access the local system.

PROCESSING-ADMISSION =
Contains information concerning the authorization of a user in the local system to perform follow-up processing.

PROCESSING-ADMISSION = *SAME
The relevant TRANSFER-ADMISSION values (see above) of the local system apply. This specification is only of value if the local TRANSFER-ADMISSION has either been explicitly supplied with USER-ID, ACCOUNT and PASSWORD or has been supplied implicitly via the transfer admission of an admission profile that contains these specifications. Otherwise the PROCESSING-ADMISSION must itself explicitly contain these specifications, either through their entry in this NCOPY command or through the use of a corresponding admission profile.

PROCESSING-ADMISSION = *NOT-SPECIFIED
Only if FTAC functionality is used can the entry for PROCESSING-ADMISSION be predefined by an FT profile. This entry must not be specified in the FT request.

PROCESSING-ADMISSION = *PARAMETERS(...)
User ID, account number and password of the user for whom the follow-up processing is to be performed. The operands in parentheses can also be used as positional operands without their keywords.

USER-IDENTIFICATION = <name 1..8>
User ID in the local system. This ID must be specified in the syntax of the local z/OS system.

ACCOUNT = *NONE
Access is possible without a password.
The default account number of the user ID specified in the USER-IDENTIFICATION is used.

ACCOUNT = <alphanum-name 1..40> / <c-string 1..40>
Account number or "accounting information" of the user in the local system. The account number must be specified in the syntax of the local system.

PASSWORD =
Password authorizing the user to access the local system.

PASSWORD = *NONE
Access is possible without a password.

PASSWORD = <alphanum-name 1..8>
Password that authorizes the user to access the local system.

Follow-up processing in the local system

A command sequence can also be input for SUCCESS-PROCESSING and FAILURE-PROCESSING. The individual commands must be separated by a semicolon. If a character string is enclosed in single or double quotes (' or ") within a command sequence, openFT will not interpret any semicolon present within this character string as a separator.

Example

```
SUCC = 'command1;command2;command3'
```

The entries for the operands SUCCESS-PROCESSING and FAILURE-PROCESSING may total up to 1000 characters. If the length limit for follow-up processing data is exceeded due to variable replacement, openFT ends the request and returns an error message.

The commands/JCL statements must comply with the z/OS syntax rules; openFT does not check the syntax of the commands and statements specified for follow-up processing, see "Syntax rules for follow-up processing on z/OS" on page 82.

Please refer to section "Follow-up processing" on page 82 for details on follow-up processing such as using variables, for example.

SUCCESS-PROCESSING =
Follow-up processing to be executed in the local system after a successful file transfer.

SUCCESS-PROCESSING = *NONE
No follow-up processing to be executed.

SUCCESS-PROCESSING = <c-string 1..1000 with-low>
z/OS commands to be executed in the local system after successful file transfer.

FAILURE-PROCESSING =
Follow-up processing to be carried out in the local system if an error is detected after setting up the link or during the file transfer.

FAILURE-PROCESSING = *NONE
No follow-up processing to be executed.

FAILURE-PROCESSING = <c-string 1..1000 with-low>
z/OS commands to be executed in the local system after unsuccessful file transfer. The same specifications are hold for FAILURE-PROCESSING as for SUCCESS-PROCESSING, see above.

LISTING =
Requests a result list in the local system. This listing is generated for the user for whom file transfer is performed. The default setting for LISTING in the local parameters is *STD.
If the result list is printed, the job initiated by openFT for this purpose causes an asynchronous end message to be sent to the user whose user ID is specified in the local TRANSFER-ADMISSION (JOB parameter NOTIFY).

LISTING = *STD
The value is taken from the FT parameter library as defined in the PARM element, LISTPARM key word. If no value is defined there then no result list is generated (same as *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:

inst.T*transfer-id*.LST (the user ID is the "first level qualifier")

Here, *inst* stands for the instance name of the current openFT instance and t*ransfer-id*. for the identification number of the FT request.

If the "transfer-id." is longer than 7 characters, it is divided into two parts in this filename as follows: instance name.Tpart1-id.Tpart2-id.LST, where "part1-id" always has a length of 7 characters and "part2-id" always has a length of 1 to 3 characters. The file is created on the same volume as non-existent receive files. The FT administrator can specify the volume for these (the same for all transfer requests). If the FT administrator has not specified a volume, the system defaults for newly created files apply (see also the section “Result lists generated by openFT (z/OS)” on page 94).

LISTING = *PARAMETERS(...)
Requests a result list in the local system. The list is created for the user for whom file transfer is performed.

OUTPUT =
Output medium.

OUTPUT = *SYSLST
The result list is printed out.

OUTPUT = *LISTFILE
openFT stores the result list under the ID specified in the local TRANSFER-ADMISSION operand. This file has the following name:

inst.Ttransfer-id.LST (the user ID is the "first level qualifier")

For more information see LISTING=*LISTFILE on page 363.

CONDITION =
Condition under which a result list should be generated.

CONDITION = *ANY
A result list is generated in every case.

CONDITION = *ON-FAILURE-ONLY
A result list is only generated when the file transfer is aborted with an error.

CODED-CHARACTER-SET =
Coding (character set) that is to be used to read or write the local file.

CODED-CHARACTER-SET = *STD
The character set used by default to read or write the local file is the character set specified globally with FTMODOPT or defined in the FT parameter library.

CODED-CHARACTER-SET = <alphanum-name 1..8>
Coding that is to be used to read or write the local file. The character set must be known in the local system.

Specifications for the remote system (REMOTE-PARAMETER)

REMOTE-PARAMETER =
Contains information about or for the remote system. This entry specifies the type of remote system. It also defines the syntax in which the remote system expects the value assignments.

REMOTE-PARAMETER = *BS2000(...)
The value assignments for the remote system are given in BS2000 syntax. The local system checks whether the specified values conform to this syntax.

REMOTE-PARAMETER = *MSP(...)
The value assignments for the remote system are in the syntax of the z/OS 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 single quotes. The single quotes within a value assignment must be doubled (e.g. PASSWORD= ’C’’ABCD’’’).

FILE-NAME =
Name of the file or the library in the remote system (send file or receive file). It must be specified in the syntax and conform to the conventions of the remote system.

REMOTE-PARAMETER=	***BS2000**	***MSP**	***ANY**
relevant for:	X	X	X
default value:	*NOT-SPECIFIED	*NOT-SPECIFIED	*NOT-SPECIFIED

openFT partners 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.

FILE-NAME = *NOT-SPECIFIED
relevant for *BS2000, *MSP and *ANY:
Only if FTAC functionality is used in the remote system can the file name be predefined, either partially or completely, in an FT profile. The file name or partial file name does not then have to be known to the request submitter. The file name may not be specified in the command.

FILE-NAME = <filename 1..54> / <filename 1..59> / <c-string 1..512 with-low>
Name of the file or preprocessing command, when receiving, or of the postprocessing command, when sending. The specifications differ for **with** and **without** pre- and postprocessing.

REMOTE-PARAMETER =	***BS2000**	***MSP**	***ANY**
FILE-NAME=	<filename 1..54> <c-string 1..512 with-low> *POSIX(NAME = <posix-pathname 1..510>	<filename 1..59> <c-string 1..512 with-low>	<c-string 1..512 with-low>

*Specifications **without** pre- or postprocessing on FILE-NAME*

Name of the file in the remote system (send or receive file):

- With *BS2000 and *MSP, this file name can be specified without a user ID if the file is cataloged under the user ID for which the file transfer is performed (TRANSFER-ADMISSION operand).
- This file name must be specified with a user ID ($userid.filename) if the file is not cataloged under the user ID for which the file transfer is performed (TRANSFER-ADMISSION operand).
- With FILE =<filename 1..59>, it is also possible to address library elements in z/OS provided that the FT product used in the partner system supports the transfer of library elements. For a BS2000 partner, you should use the structure *LIBRARY-ELEMENT.
- If you use the c-string data type for a partially qualified filename in a z/OS partner system (*MSP), you must specify it in the form C'FILE.XYZ'.
- If complete PO or PDSE data sets are to be transferred, the receive file must always be specified with the corresponding type :O: or :E: in the request, provided that the partner is still openFT on z/OS with a version < V10.
- If an FT product is used in the remote BS2000 and this product carries out the customary BS2000 extension of file names of the form *$filename* to include the standard user ID, the file name may be specified in this form. If this is not the case an error will result.

*Specifications **with** pre- and postprocessing on FILE-NAME*

If you specify a preprocessing command when receiving, the result from the preprocessing command is sent to the remote system's standard output (BS2000 systems: SYSLST; z/OS: SYSPRINT) before being transferred. You can also address the output from the preprocessing 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 systems it must take the form of a SAM-V file. On z/OS, this is a PS file with a variable block size.

If you specify a postprocessing command when sending, the transferred file is used as input for the postprocessing 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-System, 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 postprocessing, a c-string must be specified on FILE-NAME. The first character must be a pipe symbol '|', followed by the command string. If several commands are specified, they must be separated by a semicolon (';').

Example

```
    FILE-NAME=C‘|Command1;Command2;Command3; ...‘
```

The maximum length of the entire command is limited to the maximum length of the file name. You will find more detailed information on this in the examples starting on page 382.

If an error occurs while executing the individual commands , the transfer is aborted and the message FTR2206 or FTR2207 appears.

If a transfer request with preprocessing and postprocessing is to be restartable, the characters '|&' must be specified instead of '|'. For more detailed information, also see section "Preprocessing and postprocessing" on page 81.

Example

```
   FILE-NAME = C‘|&Command1;Command2;Command3; ...‘
```

FILE-NAME = *LIBRARY-ELEMENT(...)

REMOTE-PARAMETER =	***BS2000**	***MSP**	***ANY**
only relevant for:	X	[1]	X
default value:			*NOT-SPECIFIED

[1] For z/OS systems, library members must be defined with FILE-NAME=.

Specifies that a library member is to be transferred. *NOT-SPECIFIED for all three operands is invalid, because the entry would not guarantee access to a library member in the remote system.

Furthermore, the remote system must be capable of processing library members.

LIBRARY =
Name of the library in the remote system.

LIBRARY = *NOT-SPECIFIED
relevant for *BS2000 and *ANY.
Only when FTAC functionality is used in the remote system can the name of the library be predefined in an FT profile. The name of the library must not then be made known to the request submitter, nor may it be specified in the command.

LIBRARY = <filename 1..64> / <c-string 1..512 with-low>
relevant for *ANY.
Name of the library in the remote system. It must be specified in the conventions of the remote system.

LIBRARY = <filename 1..54> / <c-string 1..512 with-low>
relevant for *BS2000.
Name of the library in the remote system. It must be specified in the conventions of the remote system.

ELEMENT =
Name of the library member in the remote system.

ELEMENT = *NOT-SPECIFIED
relevant for *BS2000 and *ANY.
Only when FTAC functionality is used in the remote system can the name of the library member be predefined in an FT profile. The name of the library member must not then be made known to the request submitter, nor may it be specified in the command.

ELEMENT = <filename 1..64 without-gen-vers>(...) /
<composed-name 1..64 with-under>
relevant for *BS2000.
Name of the library member in the remote system. It must be specified in the conventions of the remote system.

ELEMENT = <c-string 1..64 with-low>(...)
relevant for *ANY.
Name of the library member in the remote system. It must be specified in the conventions of the remote system.

> **VERSION =**
> Version of the member in the remote system.
>
> **VERSION = <u>*NONE</u>**
> 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 = <u>*NONE</u>
relevant for *ANY.
The member type does not have to be specified in the remote system.

TYPE = *NOT-SPECIFIED
relevant for *BS2000 and *ANY.
Only when FTAC functionality is used in the remote system can the library member type be predefined in an FT profile. The type of the library member must not then be made known to the request submitter, nor may it be specified in the command.

TYPE = <name 1..8>
relevant for *BS2000.
Member type in the remote system. It must be specified in the conventions of the remote system.

TYPE = <c-string 1..8 with-low>
relevant for *ANY.
Member type in the remote system. It must be specified in the conventions of the remote system.

PASSWORD =

REMOTE-PARAMETER =	***BS2000**	***MSP**	***ANY**
relevant for:	X	X	X
default value:		*NONE	*NONE

Password authorizing access to the file in the remote system. The file password must be specified in the remote system’s syntax and conform to the conventions of the remote system.

If the file in the remote system is protected with a password, the password must be specified in this operand as:

– a write password for a receive file, or

- a read password for a send or receive file that is not protected by a write password but by a read password, or
- a password for the execution of a send or receive file that is protected neither by a read nor by a write password but by an execute command.

Newly-created receive files are not given a password by this operand. PASSWORD is ignored in such cases.

PASSWORD = *NONE
relevant for *BS2000, *MSP and *ANY.
Access is possible without a password.

PASSWORD = <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
relevant for *BS2000.
BS2000 file password.

PASSWORD = <alphanum-name 1..8>
relevant for *MSP.
z/OS file password.

PASSWORD = <c-string 1..64 with-low> / <x-string 1..128>
relevant for *ANY.
With *ANY, the file password must always be in inverted commas.

TRANSFER-ADMISSION =
Contains information on authorization to perform file transfers in the remote system.

REMOTE-PARAMETER =	***BS2000**	***MSP**	***ANY**
relevant for:	X	X	X
default value:	mandatory	mandatory	*NONE

TRANSFER-ADMISSION = <u>*NONE</u>
relevant for *ANY.
The remote system does not require/recognize any transfer admission.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <x-string 15..64> / <c-string 8..32 with-low>
relevant for *BS2000, *MSP and *ANY.
When FTAC functionality is used in the remote system, only the TRANSFER-ADMISSION predefined in the admission profile may be specified. The alphanumeric entries are converted internally to lowercase letters.
When entering the transfer admission in the form of a c-string you should bear in mind the notes regarding the distinction between uppercase and lowercase characters which can be found on page 40. With an z/OS partner system, you must remember that a

transfer admission defined as a c-string is stored in the profile using uppercase characters. To go with this you must also enter it in the NCOPY command, to ensure that conversion to lowercase is avoided.

TRANSFER-ADMISSION = *PARAMETERS(...)
Identification, account number and password of the user in the remote system for which the follow-up processing is to be performed. The operands in parentheses can be used as positional operands without their keywords.

REMOTE-PARAMETER =	***BS2000**	***MSP**	***ANY**
USER-IDENTIFICATION =	<alphanum-name 1..8>	<name 1..8>	<c-string 1..67 with-low>
ACCOUNT=	*NONE <alphanum-name 1..8>	*NONE <alphanum-name1..40> / <c-string 1..40>	*NONE <c-string 1..64 with-low>
PASSWORD=	*NONE <c-string 1..8> / <c-string 9..32> <x-string 1..16>	*NONE <alphanum-name 1..8>	*NONE <c-string 1..64 with-low> / <x-string 1..128>

USER-IDENTIFICATION =
relevant for *BS2000, *MSP and *ANY.
Identification of the user (user ID) in the remote system.

ACCOUNT =
relevant for *BS2000, *MSP and *ANY.
Account number of the user in the remote system.

ACCOUNT = *NONE
relevant for *BS2000, *MSP and *ANY.
The remote system does not require an account number.

PASSWORD =
relevant for *BS2000, *MSP and *ANY.
Password authorizing the user to access the remote system.

PASSWORD = *NONE
relevant for *BS2000, *MSP and *ANY.
Access is possible without a password.

PROCESSING-ADMISSION =
relevant for *BS2000, *MSP and *ANY.
Contains information about a user’s authorization to perform follow-up processing in the remote system.

i FTP partners do not support remote follow-up processing.

PROCESSING-ADMISSION = *SAME
The relevant REMOTE TRANSFER-ADMISSION values apply.

PROCESSING-ADMISSION = *NONE
relevant for *ANY.
No transfer admission is required for follow-up processing. See also the description of PROCESSING-ADMISSION=*NOT-SPECIFIED.

PROCESSING-ADMISSION = *NOT-SPECIFIED
Only if FTAC functionality is used in the remote system can the PROCESSING-ADMISSION be predefined in an FT profile. It must not then be made known to the request submitter, nor may it be specified in the command.

PROCESSING-ADMISSION = *PARAMETERS(...)
Identification, account number and password of the user in the remote system, for which the follow-up processing is to be performed. The parameters in parentheses can be used as positional operands without their keywords.

REMOTE-PARAMETER =	***BS2000**	***MSP**	***ANY**
USER-IDENTIFICATION =	<alphanum-name 1..8>	<name 1..8>	<c-string 1..67 with-low>
ACCOUNT =	*NONE <alphanum-name 1..8>	*NONE <alphanum-name1..40> / <c-string 1..40>	*NONE <c-string 1..64 with-low>
PASSWORD =	*NONE <c-string 1..32> / <x-string 1..16>	*NONE <alphanum-name 1..8>	*NONE <c-string 1..64 with-low> / <x-string 1..128>

USER-IDENTIFICATION =
Identification of the user (user ID) in the remote system.

ACCOUNT =
Account number of the user in the remote system.

ACCOUNT = *NONE
relevant for *BS2000, *MSP and *ANY.
The remote system does not require an account number.

PASSWORD =
Password authorizing the user to access the remote system.

PASSWORD = *NONE
relevant for *BS2000, *MSP and *ANY.
Access is possible without a password.

Follow-up processing in the remote system

A command sequence can also be input for SUCCESS-PROCESSING and FAILURE-PROCESSING.
The individual commands must be separated by a semicolon. If a character string is enclosed in single or double quotes (' or ") within a command sequence, openFT will not interpret any semi-colon present within this character string as a separator.

Example

```
SUCC='command1;command2;command3'
```

The entries for the operands SUCCESS-PROCESSING and FAILURE-PROCESSING may total up to 1000 characters. If the length limit for follow-up processing data is exceeded due to variable replacement, openFT ends the request and returns an error message.

The commands/JCL statements must comply with the z/OS syntax rules; openFT does not check the syntax of the commands and statements specified for follow-up processing, see "Syntax rules for follow-up processing on z/OS" on page 82.

Please refer to section "Follow-up processing" on page 82 for details on follow-up processing like using variables, for example.

SUCCESS-PROCESSING =
Follow-up processing to be executed in the remote system after a successful file transfer.

i FTP partners do not support follow-up processing.

REMOTE-PARAMETER =	***BS2000**	***MSP**	***ANY**
relevant for:	X	X	X
default value:	*NONE	*NONE	*NONE

SUCCESS-PROCESSING = *NONE
No follow-up processing is to be executed.

SUCCESS-PROCESSING = <c-string 1..1000 with-low>
Command to be executed in the remote system after a successful file transfer. It must be specified in quotes according to the syntax and conventions of the remote system.

When in the form of a c-string, follow-up processing is enclosed in quotes.

In the case of a BS2000 partner, each command begins with a slash ('/command'). It is also possible to specify a sequence of commands separated from one another by semicolons.

In the case of z/OS partners (*MSP), the comments relating to follow-up processing in the local system apply, i.e. every command starts without a backslash ('command'). You can also specify a sequence of commands separated by semicolons (see page 373).

For all other partner systems (*ANY), the rules applying to the remote system in question apply.
When entering the follow-up processing specifications, you may also use variables provided that these are supported by the remote system (see also section “Follow-up processing” on page 82).

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.

i FTP partners do not support follow-up processing.

REMOTE-PARAMETER =	***BS2000**	***MSP**	***ANY**
relevant for:	X	X	X
default value:	*NONE	*NONE	*NONE

FAILURE-PROCESSING = *NONE
No follow-up processing is to be executed.

FAILURE-PROCESSING = <c-string 1..1000 with-low>
Command to be executed in the remote system if the file transfer is aborted because of an error. It must be specified in quotes according to the syntax and conventions of the remote system.

The same specifications are valid for FAILURE-PROCESSING as for SUCCESS-PROCESSING, see above.

CODED-CHARACTER-SET=
Coding (character set) that is to be used to read or write the remote file.

CODED-CHARACTER-SET= *STD
The character set used by default to read or write the remote file is the character set defined as the default in the remote system.

CODED-CHARACTER-SET= <alphanum-name 1..8> / <c-string 1..8 with-low>
Coding (CCS) that is to be used to read or write the remote file. The character set must be known in the remote system.

Optional entries

The optional entries permit you to set special conditions for the operation and time frame of your file transfers. The optional entries deal with the type of data transfer:

- compressed (COMPRESS)
- encrypted (DATA-ENCRYPTION)
- coding of the send file (DATA-TYPE)
- write rules for the receive file (WRITE-MODE)
- maximum record length (RECORD-SIZE)
- tabulator expansion (TABULATOR)

COMPRESS =
Defines whether the data in the send file is to be transferred in compressed form.

COMPRESS = *NONE
The data in the send file is transferred uncompressed.

COMPRESS = *BYTE-REPETITION
The data in the send file is transferred in compressed form. Compression affects consecutive bytes with identical contents. If file transfer in compressed form is not possible, the data is transferred in uncompressed form.

COMPRESS = *ZIP
The data in the send file is transferred in compressed form. Compression affects consecutive bytes with identical contents. If file transfer in compressed form is not possible, the data is transferred in uncompressed form.

WRITE-MODE =
Determine how the data is to be written into the receive file. Three options are available. You can

- overwrite an already existing file in the receiving system.
- set up a new file in the receiving system. If a file with the same name already exists in the receiving system, it will not be overwritten.
- attach the transferred file to a file which already exists in the receiving system.

WRITE-MODE = *REPLACE-FILE
Overwrites the receive file from start of file. If the receive system already contains a file with this name, this file and where necessary its file attributes are overwritten. The previous contents of this file are thus completely erased. If the destination does not already exist, it is newly created.

WRITE-MODE = *NEW-FILE
Writes the receive file from start of file. If the receive system already contains a file with this name, this file is not overwritten and the send file is not transferred.

It should be noted that the receive file can already exist following the abortion of a file transfer request. It is not deleted in this case. If a new attempt is made, the request is rejected in the case of WRITE-MODE=*NEW-FILE, as the file already exists.

WRITE-MODE = *EXTEND-FILE
The receive file is extended from the end of file and written to end of file from this point. If the receive system does not yet include a file with this name, a new receive file is created. If the partner is a BS2000 system, then it depends on the system characteristic whether a request with the specification WRITE-MODE=EXTEND-FILE will be accepted or not.
The specification WRITE-MODE=*EXTEND-FILE is not permitted when transferring an entire PO or PDSE data set.

The specification WRITE-MODE=EXTEND-FILE is permitted in other cases only if:

– send file and receive file have the same record formats,
– for send files and receive files with fixed-length records the record length is the same, and
– the buffer of the receive file can accept the largest record in the send file.

If a file transfer with WRITE-MODE=EXTEND-FILE is aborted permanently, the receive file retains the contents it had at the moment the transfer was terminated.

DATA-TYPE =
Coding used for data in the send file.

DATA-TYPE = <u>*NOT-SPECIFIED</u>
For openFT partners:
The specification is interpreted in the same way as DATA-TYPE=*BINARY if the partner system is an openFT (BS2000) system and the transferred file is neither a POSIX file nor a library member. Otherwise the specification is interpreted in the same way as DATA-TYPE=*CHARACTER.
For FTAM partners:
The send file type is unknown and is defined by the send system.
In z/OS, the specification is interpreted as DATA-TYPE = *CHARACTER.

DATA-TYPE = *USER
The send file contains structured binary data of variable record length. On Unix and Windows systems, a 2-byte field specifying the record length precedes each record. The maximum record length is 32767 bytes.

DATA-TYPE = *CHARACTER(...)
The send file is transferred as a text file. The receive system stores the file in its character code as text (i.e. a code conversion is performed on the file if necessary).

DATA-TYPE = *BINARY(...)
The send file is transferred as a binary file. The receive system stores the file as it was supplied by the send system. No code conversion takes place.

TRANSPARENT =
Specifies if the file is to be converted to a transparent format.

TRANSPARENT = *NO
No transparent format should be generated. If a file in transparent format is sent to a system that supports transparent transfer, then the file is automatically set up there again with its original attributes.

TRANSPARENT = *YES
This specification is only of use to retrieve files from a partner system that supports transparent transfer. z/OS can then act as temporary storage for such files. The partner system converts the send file into a transparent format and flags it internally as a text or binary file.
If a transparent file is to be returned to the partner, this must always take the form of a binary file with TRANSPARENT=*NO. The partner system automatically recognizes that it has received a transparent file and creates the file again with its original attributes.
In the case of send requests, TRANSPARENT=*YES is ignored.

PRIORITY =
Priority with which the file transfer is initiated relative to other file transfers to the same remote system.

PRIORITY = *NORMAL
The file transfer has normal priority.

PRIORITY = *HIGH
The file transfer has high priority.
Requests with high priority executed via openFT protocols can interrupt normal priority requests for the time it takes to terminate those high priority requests. The interrupted requests are then restarted.

PRIORITY = *LOW
The file transfer has low priority.

START =
Time when the file transfer is to start. The application of the operand is accurate to approximately 5 minutes.

START = *SOON
The file transfer starts as soon as the resources required are available.

START = *EARLIEST(...)
The file transfer starts as soon as the resources required are available and not prior to the time specified. Up to this point the file transfer request is kept in a HOLD state. The date and time specified must not be further ahead than 22 days and 14 hours at the most. If the date and time specified have already passed, the file transfer is executed as if START=*SOON had been specified.

DATE =
Day when the file transfer is to be initiated.

DATE = *TODAY
The file transfer is initiated at the earliest on the day the command is issued.

DATE = *TOMORROW
The file transfer is initiated at the earliest on the day following issue of the command.

DATE = <date 8..10>
The file transfer is initiated on the calendar day specified. If the year is defined by four digits, it must be a year between 1960 and 2059. If only two digits are entered, an internal procedure extends the figure to four digits to denote a year between 1960 and 2059.

TIME = 00:00 / <time 1..8>
The file transfer is initiated at the earliest on the day following issue of the command.

CANCEL =
Specifies whether and when the file transfer is to be aborted. The application of the operand is accurate to approximately 5 minutes.

CANCEL = *NO
The file transfer is not to be deliberately aborted.

CANCEL = *AT(...)
The file transfer is to be aborted at a specific point in time.
The time specified must not

- have already passed,
- be more than 22 days and 14 hours after the specified start time,
- be before or the same as the time specified in the START operand.

DATE =
Day when the file transfer is to be aborted.

DATE = *TODAY
The file transfer is aborted on the day the command is issued.

DATE = *TOMORROW
The file transfer is aborted on the day following issue of the command.

DATE = <date 8..10>
The file transfer is aborted on the calendar day specified. If the year is defined by four digits, it must be a year between 1960 and 2059. If only two digits are entered, an internal procedure extends the figure to four digits to denote a year between 1960 and 2059.

TIME = 23:59 / <time 1..8>
The file transfer is aborted at the specified time on the chosen calendar day.

DATA-ENCRYPTION =
Determines whether or not the file transfer is to be encrypted.

DATA-ENCRYPTION = *NO
The file contents are not transmitted in encrypted form.

DATA-ENCRYPTION = *YES
The file contents are transmitted in encrypted form. If encryption is not 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 maximum record length is specified explicitly then this value is used even if the record length is known from the catalog. If a record is transferred that exceeds this maximum record size, the request is aborted with

```
FTR2087 OPENFT: Request >>1<<. File structure error >>2<<
```

RECORD-SIZE = *NOT-SPECIFIED
The maximum record length is automatically determined from the catalog.

RECORD-SIZE = <integer 1..32756>
Maximum record length of the data that is to be transferred.

RECORD-FORMAT =
Indicates how the data is transferred on a file transfer to or from a partner.

RECORD-FORMAT = *STD
The record format specification is unchanged.

RECORD-FORMAT = *FIXED
The data is transferred in fixed length records.

RECORD-FORMAT = *VARIABLE
The data is transferred in variable length records.

RECORD-FORMAT = *UNDEFINED
The record length used for data transfer is not mapped to the real system. This means that the record length used for transfer is not identical to the record length in the real file.
In the case of text files, each record is terminated with an end-of-record character both during transfer and then in the real system. Binary files are stored as bit strings in the real system.

TABULATOR =
Specifies whether tab expansion is activated.

TABULATOR = <u>*AUTO</u>
The system uses tab expansion as required.

TABULATOR = *ON
Tab expansion is activated.

TABULATOR = *OFF
Tab expansion is deactivated.

TARGET-FILE-FORMAT =
This operand allows the format of the target file to be specified.

TARGET-FILE-FORMAT = <u>*SAME</u>
The format of the target file is to be the same as that of the send file.

TARGET-FILE-FORMAT = *BLOCK-ORIENTED
The file is to be stored with a block structure. As of openFT V11.0, support is only offered for creating a block-structure file in BS2000 and in PAM format. Creation of a block-structure file in the remote system is only supported with the openFT protocol. Transfer must be performed in binary format. If the file type is specified neither in the command (DATA-TYPE) nor in the file catalog, binary transfer is automatically assumed.

The PAM file created depends on the pubset type (PAMKEY, DATA, DATA-4K). Each of the blocks is completely filled with the binary data stream received. If the data originally comes from a PAM file, the PAM keys are lost during transfer, and the file structure may be lost if the formats of the sending and receiving pubsets differ.

If openFT V10 is running on the receiving system, the file is created as a sequential file with an undefined record format. If older openFT versions are used in the receiving system, the request is rejected.

TARGET-FILE-FORMAT = *SEQUENTIAL (...)
The format of the target file is to be sequential. This also makes it possible to read block-structure files and index sequential files sequentially. The reading of PAM files and ISAM files in BS2000 is supported in openFT version 11.0:

– A PAM file is mapped to a binary sequential file with an undefined record format. The transfer is compatible with standard FTP transfer in BS2000.

– An ISAM file is mapped to the corresponding sequential format (fixed or variable record format). The contents of the ISAM keys is retained in the records, but the key positions are lost.

Specifying *SEQUENTIAL for a sequential send file has no effect.

RECORD-FORMAT =
The record format can be specified for a sequential target file.

RECORD-FORMAT = *SAME
The record format of the target file is to be the same as that of the send file.

RECORD-FORMAT = *UNDEFINED
The record format of the target file is to be undefined. The record structure of the send file is lost. At least one block is written for each transfer unit on target systems running BS2000 or z/OS. This can significantly increase the required disk storage space, for instance if the send file is made up of variable length records.

If the FT request is free of errors from the perspective of the local system, then the FT system outputs the following report as an FT request confirmation:

```
FTR0000 OPENFT: Request (&00) accepted
```

(&00) in this case, is the Identification of the FT request that assigns the local FT system to each FT request. Using this FT request ID, you can cancel the FT request (NCANCEL command), or you can get information on the status of the FT request (NSTATUS command). The FT request ID may consist of a maximum of ten decimals. You can, of course, access your FT requests, even if you do not know the FT request ID (see the information following page 400).

If the local system cannot accept the request, it issues the relevant error message (e.g. due to a syntax error in the command or because the user is not authorized to access the send or receive file). The FT messages are explained in the manual "openFT (z/OS) - Installation and Operation".

If the request has been accepted by the local system but cannot be executed, you will find the relevant error message in the result list, provided you have specified that a result list is to be created (see the LISTING parameter).

Information on the asynchronous messages issued by the local system on termination of the file transfer request is given in the section “Messages and return codes automatically issued by openFT (z/OS)” on page 92.

If neither a result list nor asynchronous messages provide information on the success or failure of the request, you can use the logging function to determine whether the request has been completed.

4.44.3 Examples of the NCOPY command

This section provides sample applications of the NCOPY command.

Further examples are available on the openFT product volume; please consult your FT administrator.

1. NCOPY command for openFT with mandatory operands only

   When the conditions on page 351 apply, the NCOPY command can be entered only with the mandatory operands.

   In the following example the file DATA is to be transferred from the local computer to the partner computer HOST001.
   The command is entered in the recommended short form.

   ```
   NCOPY TO,HOST001,(DATA),*MSP(DATA,,(MICKEY,,1313,abc))
   ```

   The long form of this command is as follows:

   ```
   NCOPY TRANSFER-DIRECTION=TO,PARTNER=HOST001,
   LOCAL-PARAMETER=(FILE-NAME=DATA),
   REMOTE-PARAMETER=*MSP(FILE-NAME=DATA,
   TRANSFER-ADMISSION=(USER-IDENTIFICATION=MICKEY,ACCOUNT=1313,PASSWORD=abc))
   ```

2. Transfer of a file with password protection cataloged under another user ID in the remote system.

   The file LIST is stored in computer HOST002 under the user ID SHIPPING and protected by the password C’XX’. The user ID SHIPPING has the account number SHIP002 and the password TOPSEC.
   The command is entered in the local z/OS system under the identifier CENTRAL.

   The example below shows both the short form and the long form of the command used to transfer the file LIST to the local system and store it there in the file *LIST.ABC*, which has not yet been created. If this file already exists, the LIST file should not be transferred.

   Recommended short form of the command:

   ```
   NCOPY FROM,HOST002,(LIST.ABC),*MSP(LIST,XX, -
            (SHIPPING,SHIP002,TOPSEC)),,NEW
   ```

   ```
   FTR0000 OPENFT: Request 31485389 accepted
   ```

   Long form of the command:

   ```
   NCOPY PARTNER=HOST002,TRANSFER-DIRECTION=FROM-PARTNER,
   LOCAL-PARAMETER=(FILE-NAME=LIST.ABC),
   REMOTE-PARAMETER=*MSP(FILE-NAME=LIST,PASSWORD=XX,
   TRANSFER-ADMISSION=(USER-IDENTIFICATION=SHIPPING,
   ACCOUNT=SHIP002,PASSWORD=TOPSEC)),
   WRITE-MODE=*NEW-FILE
   ```

   ```
   FTR0000 OPENFT: Request 16085132 accepted
   ```

3. Collection of files
   A central office has to collect the monthly reports from its 5 branch offices on the first of every month. These monthly reports are edited ready for printing in each of the branch offices and contained in a file called REPORT.*month* and are each to be transferred into a file in the central location called REPORT.*month.branch-office*. The user IDs do not contain any passwords.

   The transfer of these files is carried out with the following CLIST procedure:

```
PROC 1 MONTH
/*       PLEASE ENTER LAST MONTH FOR MONTH!                          */
NCOPY TRANS=FROM,PARTNER=BRANCH1, +
LOC=(FILE-NAME=REPORT.&MONTH..BRANCH1, +
SUCC='SEND ''REPORT.&MONTH..BRANCH1 RECEIVED'',USER(*)'), +
REM=*MSP(FILE-NAME=REPORT.&MONTH,TRANS=(CENTRAL,CENTR01))
NCOPY TRANS=FROM,PARTNER=BRANCH2, +
LOC=(FILE-NAME=REPORT.&MONTH..BRANCH2, +
SUCC='SEND ''REPORT.&MONTH..BRANCH2 RECEIVED'',USER(*)'), +
REM=*MSP(FILE-NAME=REPORT.&MONTH,TRANS=(CENTRAL,CENTR01))
NCOPY TRANS=FROM,PARTNER=BRANCH3,  +
LOC=(FILE-NAME=REPORT.&MONTH..BRANCH3,  +
SUCC='SEND ''REPORT.&MONTH..BRANCH3 RECEIVED'',USER(*)'), +
REM=*MSP(FILE-NAME=REPORT.&MONTH,TRANS=(CENTRAL,CENTR01))
NCOPY TRANS=FROM,PARTNER=BRANCH4, +
LOC=(FILE-NAME=REPORT.&MONTH..BRANCH4, +
SUCC='SEND ''REPORT.&MONTH..BRANCH4 RECEIVED'',USER(*)'), +
REM=*MSP(FILE-NAME=REPORT.&MONTH,TRANS=(CENTRAL,CENTR01))
NCOPY TRANS=FROM,PARTNER=BRANCH5, +
LOC=(FILE-NAME=REPORT.&MONTH..BRANCH5, +
SUCC='SEND ''REPORT.&MONTH..BRANCH5 RECEIVED'',USER(*)'), +
REM=*MSP(FILE-NAME=REPORT.&MONTH,TRANS=(CENTRAL,CENTR01))
END
```

   This CLIST procedure (name: *MONTH.CLIST*) is called as follows, taking the month of November as an example:

```
EX MONTH 'NOVEMBER'
FTR0000 OPENFT: Request 30436972 accepted
FTR0000 OPENFT: Request 68185709 accepted
FTR0000 OPENFT: Request 38825582 accepted
FTR0000 OPENFT: Request 31485551 accepted
FTR0000 OPENFT: Request 37777008 accepted
```

4. Distribution of files

   A central office distributes guidelines to its five branch offices.
   This guidelines are subsequently printed at the receive system by a job which is contained in the member *PRINT* of the PO file *JOB*.

   To distribute the guidelines the central office uses the the following job:

```
//CENTRAL  JOB
//NCOPY    EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
NCOPY TRANS=TO,PARTNER=BRANCH1,                                      -
LOC=(FILE-NAME=GUIDE.LINE),                                          -
REM=*MSP(FILE-NAME=GUIDE.LINE,TRANS=(BRA1),                          -
SUCC='ALLOC DSNAME(JOB(PRINT))')
NCOPY TRANS=TO,PARTNER=BRANCH2,                                      -
LOC=(FILE-NAME=GUIDE.LINE),                                          -
REM=*MSP(FILE-NAME=GUIDE.LINE,TRANS=(BRA2),                          -
SUCC='ALLOC DSNAME(JOB(PRINT))')
NCOPY TRANS=TO,PARTNER=BRANCH3,                                      -
LOC=(FILE-NAME=GUIDE.LINE),                                          -
REM=*MSP(FILE-NAME=GUIDE.LINE,TRANS=(BRA3),                          -
SUCC='ALLOC DSNAME(JOB(PRINT))')
NCOPY TRANS=TO,PARTNER=BRANCH4,                                      -
LOC=(FILE-NAME=GUIDE.LINE),                                          -
REM=*MSP(FILE-NAME=GUIDE.LINE,TRANS=(BRA4),                          -
SUCC='ALLOC DSNAME(JOB(PRINT))')
NCOPY TRANS=TO,PARTNER=BRANCH5,                                      -
LOC=(FILE-NAME=GUIDE.LINE),                                          -
REM=*MSP(FILE-NAME=GUIDE.LINE,TRANS=(BRA5),                          -
SUCC='ALLOC DSNAME(JOB(PRINT))')
/*
//
```

   This job is stored in the member GUIDEL of the PO file SHIPPING.CNTL, which is called up as follows under user ID CENTRAL:

```
SUB SHIPPING(GUIDEL)

JOB CENTRAL(JOB00136) SUBMITTED
```

5. Chaining of files

   A central office collects SAM files (e.g. transaction files) from its 3 branch offices. The files are to be concatenated and are not to be processed until all the files to be collected have been transferred. The files have the name SAM.FILE in the example and are to be stored consecutively in the file SAM.ALL.
   The user IDs do **not** contain any passwords.

   The files are transferred using the following CLIST procedure (name: *CHAIN.CLIST*):

   ```
   PROC 0
   NCOPY TRANS=FROM,PARTNER=BRANCH1,WRITE=EXT,                          -
   REM=*MSP(FILE-NAME=SAM.FILE,PASS=BRA1,TRANS=(BRA1,,)),               -
   LOC=(FILE-NAME=SAM.ALL,                                              -
   SUCC='NCOPY TRANS=FROM,PARTNER=BRANCH2,WRITE=EXT,                    -
   REM=*MSP(FILE-NAME=SAM.FILE,PASS=BRA2,TRANS=(BRA2,,)),               -
   LOC=(FILE-NAME=SAM.ALL,                                              -
   SUCC=''NCOPY TRANS=FROM,PARTNER=BRANCH3,WRITE=EXT,                   -
   LOC=(FILE-NAME=SAM.ALL,                                              -
   REM=*MSP(FILE-NAME=SAM.FILE,PASS=BRA3,TRANS=(BRA3,,))'')')
   ```

   You call the procedure as follows:

   ```
   EX CHAIN

   FTR0000 OPENFT: Request 30436727 accepted
   ```

6. File transfer between openFT (z/OS) and another FT system

   The file FILE is to be transferred to another system using openFT(z/OS). In the other system (SYS) the user ID BROOKLYN with the password 20000 is to be used. The file is to be given the name f/i/l/e in this system. The short form of the command is used:

   ```
   NCOPY TRANS=TO,PARTNER=SYS,LOC=(FILE-NAME=FILE),
   REM=*ANY(FILE-NAME='f/i/l/e',TRANS=('BROOKLYN','20000'))
   ```

   ```
   FTR0000 OPENFT: Request 15078927 accepted
   ```

   Note the comments in section “Differentiation between uppercase and lowercase letters” on page 40.

7. File transfer to BS2000 systems

   The file FILEB is to be transferred from an z/OS computer to the BS2000 computer BS2. The file is protected in the local system by password ZZZZ. The user ID has the account number ZENTR01 and is protected by password HQ1. In the BS2000 computer the file is to be called FILE, is to be protected against overwriting with the password C'XXXX' and be stored under the user ID CENTRBS2 with the account number CENTRAL2 and password C'CEN2'. The command is entered in the short form.

   ```
   NCOPY TO,BS2,
   (FILE-NAME=FILEB,PASS=ZZZZ),
   *BS2000(FILE=FILE,PASS=C'XXXX',TRANS=(CENTRBS2,CENTRAL2,C'CEN2'))
   ```

   ```
   FTR0000 OPENFT: Request 56465400 accepted
   ```

8. File transfer using openFT (Unix systems)

   The file mailbag is to be obtained from the Unix system ALFRED and transferred to the local z/OS system. The file is held by ALFRED under the user ID „flyte“ with a password of 144002 and is to be given the name NO.HURRY in the local system. An abbreviated command for this request is as follows:

   ```
   NCOPY FROM,ALFRED,(FILE-NAME=NO.HURRY),
   *ANY(FILE-NAME='mailbag',TRANS=('flyte',,'144 002'))
   ```

   ```
   FTR0000 OPENFT: Request 19458206 accepted
   ```

   Note the comments in section “Differentiation between uppercase and lowercase letters” on page 40.

   In this example, it is assumed that the file to be transferred is a text file (default value *CHAR for the parameter DATA-TYPE).

9. File transfer with follow-up processing

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

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

```
            PROC=(USER4,ACC4,PASS4), +
            SUCC='SEND ''FILE &FILX#############################+
######################## from &PNAM### received.'', USER(&USID##); +
                  //* PGRN=DONALD', +
            FAIL='SEND ''job &TRID##### from partner &PNAM### +
                  not correct.'', USER(&USID##); +
                  //* PGRN=DONALD')
```

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

If the job is successful in the local system:

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

If the job is unsuccessful in the local system:

```
//USER3N    JOB ACC3,
//          'MAC''DONALD',
//          MSGCLASS=X,
//          CLASS=7,
//          NOTIFY=USER3,
//          USER=USER3,
//          PASSWORD=PASS3,REGION=OM
//          EXEC PGM=IKJEFT01
//OPENFT    DD DSN=OPENFTQU.STD.CONN,
//             DISP=(SHR,KEEP)
//STEPLIB   DD DSN=OPENFTQU.OPENFT.NCLOAD,
//             DISP=(SHR,KEEP)
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
```

```
SEND 'request 1234567890 with USERID=USER3 not correctt.', USER(USER0)
/*
```

If the job is successful in the remote system:

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

If the job is unsuccessful in the remote system:

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

10. File transfer with follow-up processing (batch job)

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

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

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

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

11. File transfer using FTAC

    The file TURNOVER is to be transferred to the computer JACKJOHN. On this computer openFT (z/OS) is installed along with FTAC functionality for enhanced data protection and access control. An FT profile is provided in the computer JACKJOHN for the file transfer request.

    In order to be able to work with this FT profile, the TRANSFER-ADMISSION ‘FORMYDEARSTEVEN’ must be specified as transfer admission for the computer JACKJOHN. Specifying this gives direct access to the FT profile in the computer JACKJOHN. This FT profile contains the predetermined name that the file is to receive in the computer JACKJOHN and the predetermined details concerning follow-up processing (SUCCESS-PROCESSING and FAILURE-PROCESSING) in this computer. The value *SAME is defined for the PROCESSING-ADMISSION.

    The file transfer request must therefore contain the value *NOT-SPECIFIED for the file name.
    This specification corresponds to the default value and can therefore be omitted.
    The specification *NONE that is required for SUCCESS-PROCESSING and FAILURE-PROCESSING in this example also corresponds to the default value and can therefore be omitted. The default value *SAME for PROCESSING-ADMISSION is accepted by FTAC even if it is prespecified in the admission profile and can therefore **also** be omitted.

    The long form of the command for the file transfer is thus as follows:

    ```
    NCOPY TRANSFER-DIRECTION=TO,                                            -
          PARTNER=JACKJOHN,                                                 -
          LOCAL-PARAMETER=(FILE-NAME=SALES)                                 -
          REMOTE-PARAMETER=*MSP(TRANSFER-ADMISSION='FORMYDEARSTEVEN')
    ```

    There is of course a short form:

    ```
    NCOPY TO,JACKJOHN,(SALES),                                             -
    *MSP(TRANS-AD='FORMYDEARSTEVEN')
    ```

12. Local file processing between two openFT (z/OS) systems

    A list of the names of files for the local ID is to be transferred to the remote file SFA-FILE.LOCAL.

```
NCOPY                                                              -
   TRANSFER-DIRECTION=*TO-PARTNER, PARTNER=ZOSPART,                -
   LOCAL-PARAMETER=(FILE-NAME=                                     -
   C‘|LISTCAT OFILE(SYSPRINT)‘,                                    -
   TRANS-ADM=(USER=STEVEN,ACCOUNT=XXXX,PASS=TOPSEC)),              -
   REMOTE-PARAMETER=*MSP(FILE-NAME=SFA-FILE.LOCAL,                 -
   TRANSFER-ADMISSION=PROFZOSPART)

   FTR0000 OPENFT: Request 197292 accepted
```

13. Remote preprocessing between two openFT (z/OS) systems

    A list of the FT partner systems in the remote system is to be transferred to the local file INFO.ZOSPART.

```
NCOPY                                                              -
   TRANSFER-DIRECTION=*FROM-PARTNER, PARTNER=ZOSPART,              -
   LOCAL-PARAMETER=(FILE=INFO.ZOSPART),                            -
   REMOTE-PARAMETER=*MSP(FILE=                                     -
   C‘|FTSHWPTN OUT=*STDOUT‘,                                       -
   TRANSFER-ADMISSION=PROFZOSPART)

   FTR0000 OPENFT: Request 197294 accepted
```

14. FTINFO command for remote preprocessing

    You want to determine what openFT version is installed on a remote computer.

```
/TRANSFER-FILE                                                 -
/   TRANSFER-DIRECTION=*FROM-PARTNER, PARTNER=UNKNOWN,         -
/   LOCAL-PARAMETER=(FILE-NAME=FTINFO.UNKNOWN),                     -
/   REMOTE-PARAMETER=*ANY(FILE-NAME=C‘|ftinfo -csv‘,                -
/   TRANSFER-ADMISSION=C’PROFUNKNOWN’)

    FTR0000 OPENFT: Request 197296 accepted
```

    The file FTINFO.UNKNOWN then has the following content:

```
CmdUiVer;CmdTiVer;OsType;UserId;IsFtAdm;IsFtacAdm;FtLang;CcsName;Home;Limi
ted;IsAdmAdm;ProdVer;SrcVer;Inst;TimeOffset;FtScriptDir;NativeX25;SingleUs
er;Crypt
1210;0;"z/OS";"OPFTAAA";1;1;"en";"IBM1047";"OPFTAAA";*NO;0;"12.1A00";"355"
;"FT121";3600;"";*NO;*NO;*YES
```

    The output, in sequence of occurrence, has the following meaning:
    openFT V12.1 is installed on the remote system and the operating system is z/OS. FTINFO was issued under the user ID OPFTAAA which possesses both FT and FTAC administrator admissions (otherwise "0" instead of "1"). The openFT user interface in the remote system "speaks" English (otherwise "de" for German).

4.45 NSTATUS Query status of file transfer request

Note on usage

User group: FT user and FT administrator

Alias name: FTSHWREQ

Functional description

The NSTATUS command allows you to request information about FT requests. As with NCANCEL, you can specify selection criteria in order to obtain information about specific FT requests.

FT users can only obtain information about the FT requests they own.

The FT administrator can obtain information about the requests of any owner. For this purpose, he/she must enter the NSTATUS command in FT administration mode.

The owner of requests issued in the local system is the user ID under which they are submitted. The owner of requests issued in the remote system is the user ID in the local system under which the requests are executed.

The scope of information to be output can be selected. By default the following information is output by the system in response to the NSTATUS command:

- the transfer ID of the request,
- the initiator of the request (local or remote system),
- the operating status of the request (see description of operands for more details),
- the partner system,
- the transfer direction,
- the name of the file to be transferred in the local system,
- the number of bytes transferred.

By entering INFORMATION=*ALL in the NSTATUS command more information can be obtained. openFT then, in addition to the standard output, outputs the values of further operands of the transfer command that was used to issue the request. Which output parameters are displayed depends on the parameters which were specified for the request.

The complete description of all possible output parameters and values is provided in the section “Description of the long output of NSTATUS” on page 402.

The more precise your information request, the fewer irrelevant requests are output.

The specification of INFORMATION=*SUMMARY returns a small table with the number of jobs in the various request states.

Format

```
NSTATUS / FTSHWREQ

 TRANSFER-ID = *ALL / <integer 1..2147483647>
,SELECT = *OWN / *PARAMETERS(...)
    *PARAMETERS(...)
     |   OWNER-IDENTIFICATION = *OWN / *ALL / <name 1..8>
     |  ,INITIATOR = (*LOCAL, *REMOTE) / list-poss(2): *LOCAL / *REMOTE
     |  ,PARTNER = *ALL(...) / <text 1..200 with-low>
     |       *ALL(...)
     |        |   PARTNER-STATE = *ALL / *ACTIVE
     |  ,FILE-NAME = *ALL / <filename 1..59> / <c-string 1..512 with-low>
     |  ,MONJV = *NONE /
     |  ,JV-PASSWORD = *NONE
     |  ,STATE = *ALL / *SUSPEND / *LOCKED / *WAIT / *ACTIVE / *CANCELLED / *FINISHED / *HOLD
     |  ,GLOBAL-REQUEST-ID = *ALL / <alphanum-name 1..10>
,INFORMATION = *STD / *ALL / *SUMMARY
,OUTPUT = *STDERR(...) / *STDOUT(...)
    *STDERR(...) / *STDOUT(...)
     |   LAYOUT = *STD / *CSV
```

Operands

TRANSFER-ID =
Transfer ID of the FT request about which information is required.

TRANSFER-ID = *ALL
Supplies information about all the owner’s FT requests.
The FT user can only obtain information about the current requests he/she owns.
The FT administrator can obtain information about all current FT requests that access his/her system.

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. Only the FT administrator can make use of this operand unrestricted.

OWNER-IDENTIFICATION = *OWN
Provides information only on the file transfer requests in the user’s own ID.

OWNER-IDENTIFICATION = *ALL
Provides information on FT requests in all user IDs.

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 issued 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 “Specifying partner addresses” on page 76.

FILE-NAME =
FT requests that access this file in the local system as a send file or receive file. The file name or library member name must be specified exactly as it appears in the FT request. If %UNIQUE was specified, the file name generated by openFT must be entered as the selection criterion here.

FILE-NAME = *ALL
The file name is not used as a selection criterion to define the file transfer requests on which information is to be output.

FILE-NAME = <filename 1..59> / <c-string 1..512 with-low>
Name of a file. Information is required on the file transfer requests that access this file.

MONJV = *NONE
The parameter is supported for reasons of compatibility only.

JV-PASSWORD = *NONE
The parameter is supported for reasons of compatibility only.

STATE =
Selects those file transfer requests that are in the specified status. The status of a request may change in between entry of the command and information output. This is why the output may include requests that are in a state other than the one selected with STATE.

STATE = *ALL
The status of a request is not used as a selection criterion to define the file transfer requests on which information is to be output.

STATE = *SUSPEND
Requests information on those file transfer requests that are currently in SUSPEND status (= interrupted, e.g. by the command FTMODOPT STATE=*INACTIVE in the remote system or by a high-priority FT request).

STATE = *LOCKED
Requests information on FT requests that are in the LOCKED operating status (= temporarily locked as a result of a longer term resource bottleneck).

STATE = *WAIT
Requests information on those file transfer requests that are currently in WAIT status (= waiting for resources).

STATE = *ACTIVE
Requests information on those file transfer requests that are currently in ACTIVE status (= being processed).

STATE = *CANCELLED
Requests information on those file transfer requests that were canceled and are waiting for negotiation with the communications partner to be concluded. These requests are visible only to the FT administrator.

STATE = *FINISHED
Requests information on those file transfer requests that are currently in FINISHED status (= terminated or aborted, but where the user has not yet been informed).

STATE = *HOLD
Requests information on those FT requests that are currently in HOLD status (= awaiting the specified start time).

GLOBAL-REQUEST-ID =
Selects the FT requests on the basis of the global request identification.

GLOBAL-REQUEST-ID = *ALL
The global request identification is not a search criterion.

GLOBAL-REQUEST-ID = <alphanum-name 1..10>
Requests information on the FT request with a particular global request identification. The global request identification is relevant only for inbound requests of openFTpartners. It is assigned by the initiator of the request (transfer ID) and transferred to the local system.

INFORMATION =
Scope of the output.

INFORMATION = *STD
Output is summary form and contains the following information (see “Description of the short output of NSTATUS” on page 400):

- Transfer ID,
- Initiator,
- State of the request,
- Partner,
- Direction of transfer,
- Byte count,
- File or library member name in the local system.

INFORMATION = *ALL
Output is in full form. In addition to the summary form data, further information is provided on the operands used in the NCOPY command (see “Description of the long output of NSTATUS” on page 402).

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 of NSTATUS” on page 406).

OUTPUT =
Output medium.

OUTPUT = <u>*STDERR</u>(...)
Output is performed to SYSTSPRT or to SYSERR if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

LAYOUT = <u>*STD</u>
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 451).

If selection criteria are specified in the NSTATUS command and no request is found that matches all the specified criteria, the command is acknowledged with the following message:

```
FTR0504 OPENFT: No requests available for the selection criteria
```

4.45.1 Description of the short output of NSTATUS

Example 1

Information is to be output to SYSOUT on those FT requests submitted by the remote system ALFRED which require access to the file DRAISINE and are currently active. The required command is as follows:

```
NSTATUS SELECT=(INITIATOR=*REMOTE,PARTNER=ALFRED, -
FILE=DRAISINE,STATE=*ACTIVE)
```

The recommended short form of this command is as follows:

```
NSTATUS SEL=(INIT=*REM,PART-NAME=ALFRED,FILE=DRAISINE,STATE=*ACT)
```

The FT administrator must specify the operand OWNER=*ALL by SELECT if he/she is not the owner of the file DRAISINE.

The information is then output in the following format, for example:

```
TRANS-ID   INI STATE PARTNER  DIR  BYTE-COUNT  FILE-NAME
528184     REM ACT   ALFRED   TO   14760       DRAISINE
```

Description of the output columns:

`TRANS-ID:` Transfer ID of the file transfer request

`INI:` Initiator of the file transfer request : *REM* for REMOTE, *LOC* for LOCAL

`STATE:` State of the request (here *ACT* for ACTIVE), other outputs:

`SUSP` for SUSPEND,
Inbound request suspended, e.g. due to higher priority requests.

`LOCK` for LOCKED,

`WAIT` for WAIT,

`FIN` for FINISHED,

`HOLD` for HOLD

`PARTNER:` Symbolic name of the relevant partner system.

If the FT request is in the STATE=WAIT state, and there is no normal internal resource bottleneck, then the partner name is preceded by one of the following characters:

- `*` The FT administrator of the local system has locked a resource.
- `!` An attempt to set up a connection to the partner system failed (possibly because the remote system is not running, for example, or because FT has not been started there or, in the case of TCP/IP connections,

because the port specification contains *BY-TRANSPORTSYSTEM). This can also occur, if openFT has discovered an error during the internal check of transferred data integrity.

? Installation error.
The cause can be queried with the FT administrator.
Possible reasons:

- The remote system is connected to the local system via TCP/IP but the openFT connection to TCP/IP is interrupted.
- The authentication of the local or remote system has failed due to an unsuitable public key.

`DIR:` Transfer direction

`BYTE-COUNT:` Number of bytes transferred up to the last restart point (in the case of data compression this is the number of bytes of compressed data)

`FILE-NAME:` Name of the relevant file or library member in the local system

4.45.2 Description of the long output of NSTATUS

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 SYSPRINT on the FT request with transfer ID 67054. If the file transfer request was issued under the same user ID as that under which the inquiry is made, then the command is as follows:

```
NSTATUS TRANSFER-ID=67054,INFORMATION=*ALL,OUTPUT=*STDOUT
```

The recommended short form of this command is as follows:

```
NSTATUS TRANS=67054,INF=*ALL,OUT=*STDOUT
```

The information output on SYSLST then has the following format, for example:

```
TRANSFER-ID =67054       STORE  =16-12-11 14:37:18  FILESIZE=2000
   STATE    =WAIT        BYTECNT=0
   INITIATOR=LOCAL       TRANS  =TO                 PRIO    =NORM
   WRITE    =REPLACE     START  =SOON               CANCEL  =NO
   COMPRESS =NONE        DATA   =CHAR
   TRANSP   =NO          ENCRYPT=NO
   OWNER    =OPFTUID     DICHECK=NO
   PARTNER  =BS2PART
   FNC-MODE =TRANSPARENT
   PARTNER-STATE =ACT
   PARTNER-PRIO  =LOW
   LOC: FILE     =FILE.TEST
        TRANS-ADM=(OPFTUID,ACCOUNT)
        ASYN-MSG =ALL
   REM: FILE     =TEST2
        TRANS-ADM=REMOTE-PROFILE
```

Example 2 (Inbound request)

Full information is to be output to SYSPRINT on the FT request with transfer ID 67056. If the file transfer request was issued under the same user ID as that under which the inquiry is made, then the command is as follows:

```
NSTATUS TRANSFER-ID=67056,INFORMATION=*ALL,OUTPUT=*STDOUT
```

```
TRANSFER-ID =67056       STORE  =16-12-11 14:40:53  FILESIZE=40960000
   STATE    =WAIT        BYTECNT=10372320
   INITIATOR=REMOTE      TRANS  =FROM               PRIO    =
   WRITE    =REPLACE     START  =SOON               CANCEL  =NO
   COMPRESS =NONE        DATA   =CHAR               GLOB-ID =721214
   TRANSP   =NO          ENCRYPT=NO                 TABEXP  =NO
   OWNER    =OPFTUID     DICHECK=NO                 RECFORM =VARIABLE
   PARTNER  =BS2PART
   FNC-MODE =TRANSPARENT
   PARTNER-STATE =ACT
   PARTNER-PRIO  =NORM
   FILE     =TEST3
   TRANS-ADM=LAST
```

Meaning of the fields in the long output

The list below describes all fields which can occur in the long output (according to lines). Which fields are output in each particular case depends on the type and the parameters of the request.

`TRANSFER-ID:` Transfer ID of the request

`STORE:` The time at which the request was entered in the request queue

`FILESIZE:` The size of the file in bytes. If the output is flagged with a "K" on the right, the output is in kilobytes. If the output is flagged with "M", the output is in megabytes. The size is only shown here if the request has already been active. In the case of receive requests, a value is only shown here if the partner also sends that value.

`STATE`: State of the request

`BYTECNT:` Number of bytes transferred up to the last restart form (in the case of data compression in compressed form)

`INITIATOR:` Initiator of the request

`TRANS:` Transfer direction, as seen from local system

`PRIO:` Priority with which the request is to be started;
here: `NORM` for NORMAL.

`WRITE:` Specifies if or when the receive file is to be overwritten or extended

`START:` Requested start time of the request (SOON for “as soon as possible”).

`CANCEL:` Requested abortion time
(NO for “no abortion requested”)

`COMPRESS:` Specifies whether or not the file is to be transferred in compressed form

`DATA:` Type of file:

- `CHAR` for text file
- `BIN` for binary file
- `NOT-SPECIFIED`
  in TRANSFER-FILE (NCOPY), no DATA-TYPE was specified
- `USER` for user format

`GLOB-ID:` Global request identification, displayed only in the case of inbound requests from openFTpartners (INITIATOR=REMOTE). This corresponds to the request identification (=TRANSFER-ID) on the initiator system.

`TRANSP:` Specifies whether the transfer is to be done in transparent format

`ENCRYPT:` Specifies whether the file content is to be transferred in encrypted form

`TABEX:` Specifies whether the tabulator expansion is activated.
Possible values: ON, OFF, AUTO.

`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)

`RECFORM:` Record format. Possible values: UNDEFINED, VARIABLE, FIX.

`PARTNER:` Symbolic name of partner system participating in the request.
If the FT request is in the STATE=WAIT state, and there is no normal internal resource bottleneck, then the partner name is preceded by one of the following characters:

* The FT administrator of the local system has locked a resource.

! An attempt to set up a connection to the partner system failed (possibly because the remote system is not running, for example, or because FT has not been started there or, in the case of TCP/IP connections, because the port specification contains *BY-TRANSPORTSYSTEM). This can also occur, if openFT has discovered an error during the internal check of transferred data integrity.

? Installation error.
The cause can be queried with the FT administrator.
Possible reasons:

- The partner system is connected to the local system via TCP/IP but the openFT connection to TCP/IP is interrupted.
- The authentication of the local or remote system has failed due to an unsuitable public key

`FNC-MODE:` Encoding mode for remote file names and follow-up processing. Possible values:

`TRANSPARENT`
Encoding in transparent mode.

`CHAR` Encoding in character mode (only possible for inbound requests).

`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.

4.45.3 Description of the summary output of NSTATUS

You want to output information about the number of requests in each individual processing status.

```
NSTATUS INF=*SUMMARY
 ACT    WAIT    LOCK    SUSP   HOLD    FIN     TOTAL
   3       5       0       0      0      0        10
```

There are three requests in the ACTIVE condition, and five in the WAIT condition. Two requests are still in protocol handling, therefore the sum is 10.

4.45.4 Example of the FT administrator

The FT administrator requires information about **all** FT requests affecting his/her FT system. If comprehensive information on FT requests is wanted, one possible short form of the command is:

```
NSTATUS SEL=(*ALL), INF=*ALL
```

If the FT administrator only requires standard information about these FT requests, he/she may use one short form of the command as follows:

```
NSTATUS SEL=(*ALL)
```

The information is then output in the form (for example):

```
TRANS-ID   INI STATE PARTNER  DIR  BYTE-COUNT  FILE-NAME
242352178  LOC HOLD  TIGER    TO   0           'USER024.SRC'
242417736  REM ACT   JUMBO    FROM 128574      LISTING
242483296  REM SUSP  SYS435   TO   4582349     'USER832.FILE'
242548808  LOC ACT   XAS2     TO   765032      'PGM.LOAD'
242614296  LOC LOCK  TIGER    FROM 0           ASS.LIST
242679928  LOC WAIT  SYS435   TO   7776        'USER123.SRC'
242745512  LOC FIN   SIRIUS   TO   9457000     MONTH.STATS
```

If only the totals for all requests in the particular states is wanted, one possible short form of the command is:

```
NSTATUS SEL=(*ALL),INF=*SUM
```

The information is then output in the form (for example):

```
  ACT   WAIT   LOCK   SUSP   HOLD   FIN      TOTAL
    2      1      1      1      1     1          7
```

5 Program interface for the FT user

A file transfer request for an FT system can also be submitted from an application program. This function uses the OPENFT macro.

Description of the functions

This program interface has the following functions:

- Calling the openFT commands (with the exception of FTHELP) with all the parameters which are also valid at the command interface (see previous chapter).
- In order to call one of the above-mentioned commands the application uses the ASSEMBLER macro OPENFT. Via the LINK macro, FTNC calls the module NCOPY, which in turn sets up a connection to openFT, sends the command unchanged to openFT and waits for a message from it. After the message has been received from openFT, the connection is cleared down again.
- Messages generated by this module or created by openFT are stored as text strings in a buffer to be provided by the application program. For details of the format of these messages, see the manual "openFT (z/OS) - Installation and Operation".

  The application program then continues. A return code is supplied to the application program in register 15.
- Messages from openFT are always passed to the application program in a buffer. Even if the application program is running under TSO, no output is made to the terminal.
- With the aid of the LINK macro a search is made for the module NCOPY in the normal hierarchy of libraries in an IBM environment (TASKLIB, JOBLIB, link library). This module and the macro OPENFT are not designed to be reentrant. They can only be run in the address space below 16 Mb, but can process addresses below and above 16Mb.

Description of the interface

The ASSEMBLER macro OPENFT is provided for calling commands.

If an instance other than STD is to be used or if an SVC number other than 211 is used for the openFT subsystem, then the file<openft qualifier>.<inst>.CONN must be assigned at runtime via the DD name OPENFT. This file contains the connection data for openFT. The specifications in angle brackets stand for OPENFT QUALIFIER and the instance name of the instance that is to be used.

The following linkage conventions apply to these calls:

- In an IBM environment, registers 0, 1, 13, 14 and 15 have specific meanings and should not be used by the application program in any other way.
- Registers 0 and 1 are used by the above-mentioned macro for passing parameters.
- In register 13 the application program must pass the address of a save area of 72 bytes, justified on a full word boundary. This save area must lie in the address space below 16 Mb.
- Registers 14 and 15 are used by the above-mentioned macro for calling the module NCOPY. This module is loaded into the address space below 16 Mb.
- When control is returned to the application program, register 15 contains a return code.

The commands to be executed by openFT are passed as "command strings", which must have exactly the same format as if the commands were being entered at the terminal (see the chapter “openFT commands” on page 97). The commands must be entered in uppercase letters.

Messages returned by openFT are always passed to the application program as text strings in a buffer provided by the application program. For details of the format of these messages see the manual "openFT (z/OS) - Installation and Operation".

5.1 Macro OPENFT to call a user command

```
                          ⎧cmd-buff-addr ⎫      ⎧msg-buffer ⎫       ⎧1 ⎫
[ label ] OPENFT     CMD=⎨               ⎬,MSG=⎨            ⎬ ,VERS=⎨   ⎬
                          ⎩(r)           ⎭      ⎩(r)        ⎭       ⎩2 ⎭
```

label

Entry of a symbolic address for the first command in the macro expansion (optional).

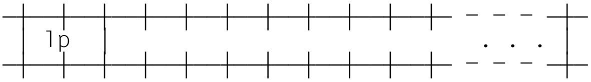
cmd-buff-addr

Address of a reference list which, in turn, also contains addresses which point, amongst others, to the command string. This list must commence on a full word boundary. Its detailed structure is described below.

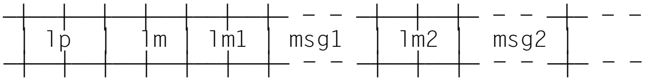
Instead of a symbolic address it is possible to specify, enclosed in brackets, the number of a register which contains this address. Registers 2 through 12 may be used for this purpose.

msg-buffer

Address of a buffer for the messages or for information passed back to the program. This buffer must commence on a full word boundary and must have the following structure when the macro OPENFT is called:



After the macro OPENFT has been executed this buffer has the following contents:



lp	Length field of 2 bytes. Contains the length of the buffer in bytes (including the length field)
lm	Length field of 2 bytes. Before a command is executed, this field must contain the value X'0000'. After the execution of a command, this field contains the length of the returned text strings in bytes, excluding the lengths of the length fields lp and lm, but including the lengths of length fields lm1, lm2, etc. If the total length of the text strings to be delivered is greater than the buffer provided, then firstly a return code of '10' (hexadecimal) is passed back to the program and the field "lm" contains the length in bytes necessary to accommodate the full text.
lm1	Length field of 2 bytes. Contains the length of the message text msg1 in bytes, excluding the length of this length field.

msg1 This field contains the first line of the message text output to the terminal as a result of executing the command.

lm2 Length field of 2 bytes. Contains the length of the message text msg2 in bytes, excluding the length of this length field.

msg2 This field contains the second line of the message text output to the terminal as a result of executing the command.

In this way all the lines of the message text are output to the buffer, until either the message text has been output in full or until the buffer length lp has been reached. The last line output to the buffer is then possibly incomplete. Although in this instance no further lines are output to the buffer, the value for "lm" continues to be accumulated, so that eventually lm contains the size which would need to be defined as the buffer size (lp) in order to accept the output in full (excluding the length of the two length fields lp and lm).

The size of the output buffer should be adapted to approximately the volume of expected data. 200 bytes are sufficient to issue an NCOPY while approximately 8KB are required for an FTSHWLOG with 100 logging records.

The format of the messages is identical to those output on the terminal, i.e. if necessary, they contain control characters. These messages are described in the manual "openFT (z/OS) - Installation and Operation".

Instead of a symbolic address it is possible to specify, in brackets, the number of a register that contains this address. Registers 2 through 12 may be used for this purpose.

VERS=

The return codes have been changed in V10.0 of openFT. This means that different message numbers are output compared with previously (openFT ≤ V9.0). The VERS parameter is used to maintain compatibility with older programs:

1 The old return codes are output, default.

2 The new return codes are output.

The VERS parameter allows you to choose between old and new return codes for output:

- If you require the old return codes to be output (default), compatibility with older programs is maintained. This means that they can be used unchanged and do not even need to be recompiled.
- If you wish to use the new return codes, you must set the parameter VERS=2 explicitly, adapt the programs and recompile them.

Structure of the reference list (cmd-buff-addr)

The parameter cmd-buff-addr points to a list, the structure of which is described below.

Transfer of a user command

In the case where a user command is transferred via the macro OPENFT, the parameter cmd-buff-addr contains the address of a list which is four full words in length and corresponds to the "command processor parameter list" (CPPL) in TSO:

```
                    ┌───┬───┬───┬───┐
cmd-buff-addr ==>   │   CPPLCBUF    │
                    ├───┼───┼───┼───┤
                    │  Nullpointer  │
                    ├───┼───┼───┼───┤
                    │  Nullpointer  │
                    ├───┼───┼───┼───┤
                    │  Nullpointer  │
                    └───┴───┴───┴───┘
```

This field CPPLCBUF contains the address of the command buffer. The command buffer to which the address in field CPPLCBUF points must commence on a full word boundary and have exactly the same structure as for a TSO command processor.

```
              ┌───┬───┬───┬───┬───┬───┬───┬───┬───┬── ─ ─ ─┬──
CPPLCBUF ==>  │  lng  │  off  │    command string     . . .  │
              └───┴───┴───┴───┴───┴───┴───┴───┴───┴── ─ ─ ─┴──
```

lng Length field of 2 bytes. This field contains the length of the command string plus the length of fields lng and off (thus, command string + 4).

off Offset field of 2 bytes. This field contains the offset of the first parameter relative to the start of the command string (not relative to the start of this buffer).

command string
This field contains the command to be executed; it is held as a character string. The length of this string is limited to a maximum of 2000 characters. The commands are described in the chapter “openFT commands” on page 97.

A return code is sent to the application program in register 15 (right-aligned). Return code (hexadecimal) in R15:

'00' Normal termination

'04' Warning

'0C' Syntax error, access conditions invalid

'10' The buffer was too small for the return message, the text was truncated.

'12' The resources (e.g. VTAM connection) are occupied.

Example of calling a user command

The following NCOPY command is to be executed (using the new return codes):

```
NCOPY  TRANS=TO,PARTNER=BS2,LOC=(FILE=DAT,TRANS=(BERT,A1234,PASSWORD)), -
       REM=*BS2000(INFO,TRANS=(FRED,4567ABC,COVERT))
```

The following ASSEMBLER code is required to effect it:

```
          LA    13,SAVAREA           SET UP SAVE AREA
          OPENFT CMD=CMDBUFF,MSG=MSGBUFF,VERS=2
          .
          .
CMDBUFF   DC    A(CBUFF)             ADDR OF COMMAND BUFFER
          DC    A(0)                 NOT USED
          DC    A(0)                 NOT USED
          DC    A(0)                 NOT USED
          SPACE
CBUFF     DS    0F                   COMMAND BUFFER
          DC    Y(CLNG)              LENGTH FIELD
          DC    Y(L'CNAME)           OFFSET TO 1ST. PARAMETER
CNAME     DC    C'NCOPY   '          COMMAND NAME
          DC    C'TRANS=TO'          1ST. PARAMETER
          DC    C',PARTNER=BS2'      2ND. PARAMETER
          DC    C',LOC=('            LOCAL PARAMETER
          DC    C'FILE=DAT'          FILE NAME
          DC    C',TRANS=('
          DC    C'BERT,A1234,PASSWORD'  TRANSFER ADMISSION
          DC    C')),'
          DC    C'REM=*BS2000('      REMOTE PARAMETER
          DC    C'INFO'              FILE NAME
          DC    C',TRANS=('
          DC    C'FRED,4567ABC,''COVERT'''  TRANSFER ADM.
          DC    C'))'
CBFFEND   EQU   *
CLNG      EQU   CBFFEND-CBUFF        BUFFER LENGTH
          SPACE
MSGBUFF   DS    0F                   MESSAGE BUFFER
          DC    Y(MSGLNG)            BUFFER LENGTH
          DC    X'0000'
          DS    CL200                MESSAGE TEXT
MSGEND    EQU   *
MSGLNG    EQU   MSGEND-MSGBUFF
          SPACE
SAVAREA   DC    18F'0'               SAVE AREA
```

6 What to do if ...

The error FTR2047 is displayed as a direct response to 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 FTSHWLOG␣ (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 FTHELP xxxx (xxxx=RC)

The error FTR2169in 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 z/OS partners: FTSHWLOG␣(RECORD-TYPE=(FT=NONE)),NUMBER=n and FTHELP xxxx With Unix or Windows partners: ftshwl␣-rt=c and fthelp␣xxxx (xxxx=RC from logging record)

The error FTRxxxx is displayed as a direct response to NCOPY

Possible cause	Suggested action
Error in the local system	Refer to 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)	Refer to manual
Exception with the error: FTR0035 `File locked to prevent multiple access.` Both the local and the remote file may be locked.	

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.
Follow-up processing is required to access files that do not exist or that cannot be accessed with the PROCESSING-ADMISSION	Check the existence of the required files and their access control settings, if necessary correct PROCESSING-ADMISSION
JES rejects follow-up processing because a required "account number" was not specified in the PROCESSING-ADMISSION	Enter the "account number" in the PROCESSING-ADMISSION
The instance-specific job TSOJOB or JCLJOB is incorrectly structured	Ask the FT administrator if there are fundamental problems with follow-up processing

Problems during the execution of preprocessing or postprocessing commands and ftexec calls

Possible cause	Suggested action
Error in the pre- or postprocessing command or a space character after a semicolon, if several pre-/postprocessing commands were specified.	Check the pre-/postprocessing commands.

Remote follow-up processing in event of error (FAILURE PROCESSING) is not executed

Possible cause	Suggested action
The file transfer command 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 NCOPY command	Enter data at the system
Message overlooked, or a LOGOFF/ LOGON command was entered after the NCOPY command	View logging records
File transfer not yet terminated	Use NSTATUS to obtain information on the FT request
The initiator is not identical to the user ID in the TRANSFER-ADMISSION	Log on under the user ID specified in the TRANSFER-ADMISSION or ask its user whether a message has been output
The user has disabled message output with PROFILE NOINTERCOM	Output the message with LISTBC MAIL; for subsequent requests enable message output with PROFILE INTERCOM
	Wait

A 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 NCANCEL with FORCE-CANCELLATION=*YES.

Despite issuing the NCANCEL command, the request has been executed

Possible cause	Suggested action
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 onthe DASD volume.	Discuss the problem with the system administrator.

No information on a TNCOPY request

Possible cause	Suggested action
	Check using FTSHWLOG␣NUMBER=n

No result list

Possible cause	Suggested action
Specify LIST=*NONE (default value)	
Printout lost	

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
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 FTSHWPTN)	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 FTSHWPTN)	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

6.1 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:

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 Windows partners, provided the file name is syntactically allowed there.

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.

How do I determine which FT requests have succeeded and which ones have failed?

The logging records output by:

```
FTSHWLOG
```

shows you the result of the last transfer.

```
/SHOW-FT-LOG ,n
```

```
FTSHWLOG ,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:

```
FTSHWLOG (REC-TYPE=(,N)),n
```

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 NCOPY commands 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.

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?

You have entered an NCOPY command that was accepted with FTR0000. It was then aborted with FTR2169.

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 z/OS with `FTSHWLOG [,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 graphical user interface; 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 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 (only under BS2000 and z/OS) and password:
Operand TRANSFER-ADMISSION=(user-id,account,password).

A simpler method is to use a special authorization exclusively for the file transfer. This is named FTAC transfer admission or shortly transfer admission (TRANSFER-ADMISSION=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 z/OS: `FTCREPRF name,,transAdm`
- In a 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.

Under BS2000 and z/OS, admission profiles can be set up only on systems with openFT-AC.

6.2 Reporting errors

If, despite taking every precaution, an error occurs that you are unable to resolve with the help of the above hints or by following the action suggested for the error message involved, please consult the FT administrator. You will make it easier for the FT administrator to perform diagnostics by providing the following documentation, where applicable:

- a precise description of the circumstances in which the error occurred and whether it can be reproduced,
- a printout of the command in which the error occurred,
- a printout of the error messages which were issued,
- if available, the result list for the request in which the error occurred (if necessary, repeat the request for the purpose of obtaining a result list, see the LISTING parameter in the NCOPY command).

7 CSV output

The output format for all commands corresponds to the following rules:

- Each record is output in a separate line. A record contains all the information to be displayed on an object.
- The first line is a header and contains the field names of the respective columns. **Only the field names are guaranteed, not the order of fields in the record.** In other words, the order of columns is determined by the order of the field names in the header line.
- Two tables, with their own respective headers, are output sequentially for the command FTSHWENV. If one of the tables is empty, the corresponding header is also dropped.
- Individual fields within an output line are delimited by a semicolon “;”.

The following data types are differentiated in the output:

- Number

  Integer

- String

  Since ";" is a metacharacter in the CSV output, any text that contains ";" is enclosed in double quotes ("). Double quotes within a text field are doubled in order to differentiate them from text delimiters. When imported into a program, the doubled quotes are automatically removed and the text delimiters removed. Keywords are output in uppercase with a leading asterisk (*) and are not enclosed in double quotes.

- Date

  The date and time are output in the form yyyy-mm-dd hh:mm:ss. In some cases, only the short form yyyy-mm-dd is ouput, i.e. the date alone.

- Time

  The time is output in the form yyyy-mm-dd hh:mm:ss or only hh:mm:ss.

Some of the fields in this command output are irrelevant for openFT for z/OS, but they appear nonetheless for reasons of compatibility with other openFT products (e.g. ElemName, ElemPrefix etc. in the output of FTSHWPRF).

7.1 FTSHW

The following table indicates the CSV output format for file attributes.

The values that are marked by an “x” in the **Std** column are also output if INF=*STD is specified. In the case of INF=*NAMES-ONLY, only the FileName column is output.

The **Parameter** column indicates the name of the output parameter in the case of detailed output, see page 265.

Column	Type	Values and Meaning	Parameter	Std
FileName	String	File name or directory name enclosed in double quotes / *NSPEC	FILENAME	x
StorageAccount	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 read / *NSPEC	REA DATE	
AtmIdentity	String	Identity of the last user of the file (modification of file attributes) enclosed in double quotes / *NSPEC	ATM name	
AtmTime	Date	Time at which the file attributes were last modified / *NSPEC	ATM DATE	
FileType	String	*BIN / *DIR / *TEXT / *NONE / *NSPEC File type	file type	x
CharSet	String	*VISIBLE / *IA5 / *GRAPHIC / *GENERAL / *NONE / *NSPEC Character set for the text file if FileType=*TEXT, in the case of another FileType, this is *NONE or *NSPEC	CHARACTERSET	
RecFormat	String	*VAR / *FIX / *NSIG / *NSPEC Record format	RECORD-FORMAT	
RecSize	Number	1... 65535 / *NSPEC Maximum length of the records	RECORD-SIZE	
FileAvail	String	*IMMEDIATE / *DEFERRED / *NSPEC File avaliability	FILE-AVAILABILITY	

Column	**Type**	**Values and Meaning**	**Parameter**	**Std**
AccessRights	String	nnnnnnnnnnn / *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	

Example

```
FTSHW ZOSMCH01,FILE3,,TRANSADM,INF=*STD,OUT=*STDOUT(*CSV)
```

```
FileName;StorageAccount;CreIdentity;ModTime;FileType;AccessRights;FileSize
"FILE3";*NSPEC;"MISTERX";2017-01-19 12:39:47;*NSPEC;r-pxeacd---;2048
```

7.2 FTSHWADS

The following table indicates the CSV output format of an admission set.

The **Parameter** column contains the name of the output parameter during normal output, see page 270.

Column	Type	Values and Meaning	Parameter
UserId	String	User ID, enclosed in double quotes / *STD *STD means standard 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 standard 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 standard 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 standard 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 standard 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 standard 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 standard 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 standard 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 standard 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 standard 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 standard 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 standard 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 standard 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 standard admission set. In the normal output, *YES corresponds to an asterisk (*) after the value.

7.3 FTSHWENV

The command FTSHWENV sequentially displays the objects contained in an FTAC export file in a format that corresponds to the output of the FTSHWADS () and FTSHWPRF () commands.

7.4 FTSHWKEY

The table below indicates the CSV format for the output of the properties of the RSA keys.

The **Parameter** column contains the name of the output parameter during normal output, see page 278.

Column	**Type**	**Values and Meaning**	**Parameter**
Reference	Number	Key reference	KEY-REF
Identification	String	Identification of the partner enclosed in double quotes / *OWN *OWN means the private key for the user’s own instance	IDENTIFICATION
PartnerName	String	Name of partner / *OWN *OWN means the private key for the user’s own instance	PARTNER
CreDate	Date	Date on which the key was generated	CRE-DATE
ExpDate	String	Date on which the key expires / *NONE	EXP-DATE
Expired	String	*YES / *NO Key has expired / not expired	EXP-DATE (EXPIRED)
KeyLen	Number	768 / 1024 / 2048 Key length in bits	KEY-LEN
AuthLev	Number	1 / 2 Authentication level	AUTHL

7.5 FTSHWLOG

The following table indicates the CSV output format of a log record if the INF=*LOGGING-FILES has not been specified. If INF=*LOGGING-FILES is specified then the output has a different format, see page 434.

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 402.

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 of the initiator / *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 admission profile enclosed in double quotes / *NONE	PROFILE	
ResultProcess	String	*STARTED / *NOT-STARTED / *NONE Status of follow-up processing	PCMD	
StartTime	Date	Start time of transfer	STARTTIME	
TransId	Number	Number of transfer request	TRANS-ID	

Column	Type	Values and Meaning	Parameter	Std
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 queue	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	
RsaKeyLen	Number	768 / 1024 / 2048 / empty Length of the RSA key used for the encryption 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	

Column	Type	Values and Meaning	Parameter	Std
Routing	String	Routing information enclosed in double quotes / empty	ROUTING	
AdmCmd	String	Administration command 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	
FileNameCMode	String	*TRANSPARENT / *CHAR Encoding mode for file names	FNC-CODE	
FileNameCcs	String	Name of the character set enclosed in double quotes (FileNameCMode=*CHAR) / empty	FNCCS	
PtnrAddr	String	Address of the partner system in the case of inbound FT requests	PTNR-ADDR	

CSV output on INF=*LOGGING-FILES

If the option INF=*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)

7.6 FTSHWMON

The following table shows the CSV output format for the monitoring values for openFT operation if all the monitoring values are output (NAME=*ALL,INF=*VALUES(..)).

If DATA=*RAW is specified, the duration values are not output (*Duxxx*, see footnote).

The default values are marked with "x" in the **Std** column. These are output if INF=*STD is specified.

For a detailed description of the monitoring values, refer to the section “Description of the monitoring values” on page 304.

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 time	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	String	*ALL / *NONE / OPENFT / FTP		Partner type selected	x
ReqSel	String	*ALL / ONLY-SYNC / ONLY-ASYNC / ONLY-LOCAL / ONLY-REMOTE		Request type selected	x
Data	String	FORM	RAW	Output format (prepared / 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 Inbound 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 time 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	
StCLim	Number	Value currently set		Maximum number of connections established for asynchronous requests.	x

Column	Type	Values prepared	Values not prepared	Meaning	Std
StCAct	Percent	Share of StCLim in %	Current number	Number of occupied connections for asynchronous requests	x
StRqLim	Number	Value currently set		Maximum number of asynchronous requests in request management	x
StRqAct	Percent	Share of StRqLim in %	Current number	Entries occupied in request management	x
StOftr	BOOL	1 / 0		openFT Protocol activated / deactivated	x
StFtmr	BOOL	1 / 0		FTAM Protocol activated / deactivated	x
StFtpr	BOOL	1 / 0		FTP Protocol activated / deactivated	x
StTrcr	BOOL	1 / 0		Trace activated / deactivated	

[1] is not output with DATA=*RAW

[2] number is the measured value multiplied by 100 (e.g. output 225 corresponds to value 2.25)

Examples

```
FTSHWMON NAME=*ALL,OUTPUT=*STDOUT(*CSV)
```

```
CurrTime;MonOn;PartnerSel;ReqSel;Data;ThNetbTtl;ThNetbSnd;ThNetbRcv;ThNetbTxt
;ThNetbBin;ThDiskTtl;ThDiskSnd;ThDiskRcv;ThDiskTxt;ThDiskBin;ThRqto;ThRqft;Th
Rqfm;ThSuct;ThAbrt;ThIntr;ThUsrf;ThFoll;ThCosu;ThCofl;ThCobr;DuRqtlOut;DuRqtl
Inb;DuRqftOut;DuRqftInb;DuRqfmOut;DuRqfmInb;DuRqesOut;DuDnscOut;DuDnscInb;DuC
onnOut;DuOpenOut;DuOpenInb;DuClosOut;DuClosInb;DuUsrcOut;DuUsrcInb;StRqas;StR
qaa;StRqwt;StRqhd;StRqsp;StRqlk;StRqfi;StCLim;StCAct;StRqLim;StRqAct;StOftr;S
tFtmr;StFtpr;StTrcr
```

```
2017-03-13 10:44:24;2017-03-13 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;5129;0;5129;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
```

7.7 FTSHWOPT

The following table indicates the CSV output format of the operating parameters.

The **Parameter** column contains the name of the output parameter during normal output, see page 315. 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: Identification / transport address	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	
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
SnmpTrapsSubsystem-State	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
ConsTrapsTransSucc	String	*ON / *OFF Console traps on successfully terminated requests activated / deactivated	TRAP, line CONS TRANS-SUCC

Column	Type	Values and Meaning	Parameter
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 / *MODIFY Scope of FTAC logging	FTAC-LOG
Trace	String	*ON / *OFF Trace function activated / deactivated	FUNCT, line TRACE SWITCH
TraceSelp	String	*ALL / OPENFT / FTP / ADM / empty [1] Trace selection based on partner type	FUNCT, line TRACE PARTNER-SELECTION
TraceSelr	String	*ALL / ONLY-SYNC / ONLY-ASYNC / ONLY-LOCAL / ONLY-REMOTE [1] Trace selection based on request type	FUNCT, line TRACE REQUEST-SELECTION
TraceOpt	String	*NO-BULK-DATA / *NONE Minimum trace / no trace options	FUNCT, line TRACE OPTIONS
KeyLen	Number	768 / 1024 / 2048 RSA key length in bit	RSA-PROP
CcsName	String	empty	---
AppEntTitle	String	*YES Not relevant on z/OS	---
StatName	String	$FJAM	
SysName	String	Name of the local system	
FtStarted	String	*YES / *NO openFT started / not started	STARTED
openftAppl	String	*STD / port number Port number of the local openFT server	OPENFT-APPL
ftamAppl	String	*NONE	FTAM-APPL
FtpPort	Number	Port number Port number of the local FTP server	FTP-PORT
ftpDPort	Number	Value / empty (internal function)	---
ftstdPort	String	*STD / port number Default port for dynamic partners	---
DynPartner	String	*ON / *OFF Dynamic partner entries activated / deactivated	DYN-PART
ConTimeout	Number	Value (internal function)	---
ChkpTime	Number	Value (internal function)	---

Column	Type	Values and Meaning	Parameter
Monitoring	String	*ON / *OFF Monitoring data activated / deactivated	FUNCT, line MONITOR SWITCH
MonSelp	String	*ALL / OPENFT / FTP / empty [1] Selection based on type of partner system	FUNCT, line MONITOR PARTNER-SELECTION
MonSelr	String	*ALL / ONLY-SYNC / ONLY-ASYNC / ONLY-LOCAL / ONLY-REMOTE [1] Selection based on type of request	FUNCT, line MONITOR REQUEST-SELECTION
AdmTrapServer	String	Name of the ADM-TRAP server / *NONE	ADM-TRAP-SERVER
AdmTrapsFtState	String	*ON / *OFF ADM traps on asynchronous server status change activated / deactivated	TRAP, line ADM FT-STATE
AdmTrapsPartnerState	String	*ON / *OFF ADM traps on partner status change activated / deactivated	TRAP, line ADM PART-STATE
AdmTrapsPartnerUnreach	String	*ON / *OFF ADM traps on unreachable partner systems activated / deactivated	TRAP, line ADM PART-UNREA
AdmTrapsReqQueueState	String	*ON / *OFF ADM traps on request management status change activated / deactivated	TRAP, line ADM RQ-STATE
AdmTrapsTransSucc	String	*ON / *OFF ADM traps on successfully terminated requests activated / deactivated	TRAP, line ADM TRANS-SUCC
AdmTrapsTransFail	String	*ON / *OFF ADM traps on failed requests activated / deactivated	TRAP, line ADM TRANS-FAIL
AdminConnLim	String	Maximum number of administration connections	ADM-CLIM
AdmPort	String	Port number / *NONE Port number for remote administration	ADM-PORT
OpenftApplState	String	*ACTIVE / *INACT / *DISABLED / *NAVAIL Status of the openFT server	OPENFT-APPL, 2nd line
FtamApplState	String	*NAVAIL Status of the FTAM server	FTAM-APPL, 2nd line
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

Column	Type	Values and Meaning	Parameter
CentralAdminServer	String	*NO	---
ActiveAppl	String	*ALL / *NONE / OPENFT / FTP / ADM[1] active servers	see 2nd line of OPENFT-APPL, FTAM-APPL, FTP-PORT, ADM-PORT
UseCmx	String	*NO	---
TraceOptLowerLayers	String	*OFF	---
EncMandIn	String	*YES / *NO Inbound encryption activated / deactivated	ENC-MAND (IN)
EncMandOut	String	*YES / *NO Outbound encryption activated / deactivated	ENC-MAND (OUT)
DelLog	String	*ON / *OFF Automatic deletion of log records activated / deactivated	DEL-LOG
DelLogRetpd	Number	Minimum age, in days, of the log records to be deleted. 0 means current day.	RETPD
DelLogRepeat	String	*MONTHLY / *WEEKLY / *DAILY Repeat interval for deletion of log records.	DEL-LOG ON
DelLogDay	Number	1..31 / 1..7 / 0 Day on which deletion is to be repeated. In 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
OutboundRecovery	String	*ON / *OFF Restart function for outbound requests	RECOVERY
InboundRecovery	String	*ON / *OFF Restart function for inbound requests	RECOVERY
ApplicationEntityTitle	String	*NSPEC	---
RSAMinimum	Number	0 / 768 / 1024 / 2048 Minimum length of the RSA key	RSA-MIN
FileNameCcs	String	empty	---
AESMinimum	String	*NONE / 128 / 256 Minimum length of AES key	AES-MIN

[1] Combinations of multiple values are also possible (not with *ALL or *NONE)

7.8 FTSHWPRF

The following table indicates the CSV output format of an admission profile.

The values that are marked by an “x” in the **Std** column are also output if INF=*ONLY-NAMES is specified.

The **Parameter** column contains the name of the output parameter during long output, see also page 324.

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 Inbound 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 unrestricted	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	Follow-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)	

7.9 FTSHWPTN

The following table indicates the CSV output format of a partner in the partner list.

The **Parameter** column contains the name of the output parameter during long output, see page 330.

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
SessRout	String	Routing information enclosed in double quotes / *ID / empty *ID means routing information same as identification	ROUTING

Column	Type	Values and Meaning	Parameter
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
OutboundRecovery	String	*FTOPT / *ON / *OFF Partner-specific restart function for asynchronous outbound requests	RECOV
ForeignPartner	String	*NO (internal function)	---
ScopeId	String	empty	---

7.10 FTSHWRGE

The following table indicates the CSV output format of partners.

The **Parameter** column contains the name of the output parameter during normal output, see page 335.

Column	Type	Values and Meaning	Parameter
SecLev	Number	Security level	SECLEV
PartnerName	String	Partner name	PARTNER-NAME

7.11 NSTATUS

The following table indicates the CSV output format of a request.

Short output is also possible with NSTATUS, see page 454.

The **Parameter** column contains the name of the output parameter during long output, see page 402.

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
DiCheck	String	*YES / *NO Data integrity is checked / is not checked	DICHECK
Write	String	*REPL / *EXT / *NEW Write rules	WRITE

Column	Type	Values and Meaning	Parameter
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 Data type	DATA
Transp	String	*YES / *NO Transfer transparent / not transparent	TRANSP
LocTransAdmId	String	User ID for accessing the local system, enclosed in double quotes / *NONE	LOC: TRANS-ADM (USER)
LocTransAdmAcc	String	Account number for the local system / *NONE	LOC: TRANS-ADM=(...account)
LocProfile	String	Name of the admission profile for accessing the local system enclosed in double quotes / *NONE	LOC: TRANS-ADM=(profile)
LocProcAdmId	String	Transfer admission for follow-up processing in the local system enclosed in double quotes / *NONE	LOC: PROC-ADM=(user...)
LocProcAdmAcc	String	Account number for follow-up processing in the local system / *NONE	LOC: PROC-ADM=(...account)
LocSuccProc	String	Local follow-up processing on success, enclosed in double quotes / *NONE / empty	LOC: SUCC-PROC
LocFailProc	String	Local follow-up processing on error, enclosed in double quotes / *NONE / empty	LOC: FAIL-PROC
LocListing	String	*SYSLST / *LISTFILE / *NONE Result list in the local system	LOC: LIST
LocMonjv	String		
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 quotes / *STD / *NONE	REM: VERSION

Column	**Type**	**Values and Meaning**	**Parameter**
RemTransAdmId	String	User ID in the remote system enclosed in double quotes / *NONE	REM: TRANS-ADM=(user-id,...)
RemTransAdmAcc	String	Account number in the remote system enclosed in double quotes / empty	REM: TRANS-ADM=(...,account)
RemTransAdmAccount[1]	String	Account number in the remote system enclosed in double quotes / empty	REM: TRANS-ADM=(...,account)
RemProfile	String	*YES / *NONE *YES means access via FTAC admission profile	REM: TRANS-ADM=REMOTE-PROFILE
RemProcAdmId	String	Transfer admission for follow-up processing in the remote system enclosed in double quotes / *NONE	REM: PROC-ADM=(user-id,...)
RemProcAdmAcc	String	Account number for follow-up processing in the remote system enclosed in double quotes / *NONE	REM: PROC-ADM=(...,account)
RemSuccProc	String	Remote follow-up processing on success, enclosed in double quotes / *NONE / empty	REM: SUCC-PROC
RemFailProc	String	Remote follow-up processing on error, enclosed in double quotes / *NONE / empty	REM: FAIL-PROC
RemCcsn	String	Name of the character set used in the remote system, enclosed in double quotes / *STD	REM: CCSN
FileSize	Number	Size of the file in bytes / empty	FILESIZE
RecSize	Number	Maximum record size in bytes / empty	RECSIZE
RecFormat	String	*STD / *VARIABLE / *FIX / *UNDEFINED Record format	RECFORM
StoreTime	Date	Time at which the request was entered in the request queue	STORE
ExpEndTime	Date	empty	---
TranspMode	String	*YES / *NO Transfer transparent / not transparent	TRANSP
DataEncrypt	String	*YES / *NO User data transferred encrypted / unencrypted	ENCRYPT
TabExp	String	*AUTO / *YES / *NO Tabulator expansion	TABEXP
Mail	String	*ALL / *FAIL / *NO Result messages	LOC: MAIL
DiagCode	String	empty	---
FileAvail	String	*NSPEC	---

Column	Type	Values and Meaning	Parameter
StorageAccount	String	empty	---
AccessRights	String	empty	---
LegalQualif	String	empty	---
PartnerPrio	String	*LOW / *NORM / *HIGH Partner priority	PARTNER-PRIO
TargetFileForm	String	*STD / *BLOCK / *SEQ File format in the target system	TARGFORM
TargetRecForm	String	*STD / *UNDEFINED Record format in the target system	TRECFRM
Protection	String	*STD / *SAME Transfer of protection attributes	PROTECT
GlobReqId	Number	Global request identification For locally issued requests, same as request ID; for globally issued requests, same as the request ID in the initiating system	TRANSFER-ID or GLOB-ID
FNCMode	String	*TRANSPARENT / *CHAR Encoding mode for remote file names and follow-up processing	FNC-MODE

[1] RemTransAdmAcc and RemTransAdmAccount have the same meaning and the same content. For reasons of compatibility, both parameters are present in the CSV output.

Short output from NSTATUS in CSV format

INF=*SUMMARY outputs a table with two rows indicating the number of requests that have the corresponding status, see also page 400.

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

Index