English

FUJITSU Software

openFT (z/OS) V12.1

Installation and Operation

System Administrator 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

The openFT product range comprises the following products:

FUJITSU Software openFT (Unix systems) is the file transfer product for systems with a Unix based operating system.

FUJITSU Software openFT (Windows) is the file transfer product for Microsoft's Windows systems.

FUJITSU Software openFT (BS2000) is the file transfer product for computers using the operating system BS2000.

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.

openFT allows the use of the following transport protocols:

- TCP/IP
- ISO TP0/2 (not on z/OS)
- ISO TP4 (not on z/OS)
- SNA (only on z/OS)

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 BS2000 systems and on z/OS, FTAC is provided by the add-on product openFT-AC.

- openFT-FTAM (not available on z/OS):

  openFT supports the FTAM file transfer protocol (File Transfer Access and Management) standardized by ISO (International Organization for Standardization). This makes it possible to interconnect with even more systems from other vendors whose file transfer products support the same standard.

- 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 FT administrators and FTAC administrators, who want to install and to start up openFT on a z/OS.

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.

. 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.


indicates notes.


Indicates warnings.

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 Installation and initial operation

This chapter describes

- how to change to a higher openFT version
- how to generate the data communication system for openFT,
- the general requirements that need to be observed for openFT operation (e.g. assignment of privileged for openFT and the protection of openFT administration files),
- the installation of openFT and of the optional delivery units openFT-FTP, openFT-AC and openFT-CR.
- the initial operation including the configuration and administration tasks that need to be completed before openFT is run.
- the configuration tasks associated with data security (with or without FTAC).

For information on the hardware and software requirements for openFT (z/OS) and connections to partner systems, please refer to the release notice.

2.1 Change of version

For openFT there is no update installation in the conventional sense but only the possibility to perform a new installation. It is therefore important to back up the configuration data.

Below you will find step-by-step instruction on the procedure for changing version.

1. Use the tools available in openFT to transfer the files of a new openFT version from the product CD to the z/OS computer, see section “Transferring files from CD to the z/OS computer and unpacking them” on page 42.
2. Use the NSTATUS command to check whether requests are still present in the request queue.
   If necessary, empty the request queue (<openft qualifier>.<inst>.SYSRQF) before shutting down the old version of openFT since it is **not** possible to take the request queue over into the new version.
3. Shut down openFT using the FTSTOP command.
4. Back up the configuration data listed below as follows (you do not need to back up other files created by openFT because openFT recreates these files):

   - Use FTSHWOPT to back up the operating parameter settings. These are the entries in the data set <openft qualifier>.<inst>.SYSOPF.

     Example

     The output from the FTSHWOPT command is converted to the correct format using LAYOUT=*ZOS-PROC and redirected to a file with the name OPTZOS.CLIST.

     ```
     READY
     FREE DDNAME(SYSPRINT)
     READY
     ALLOC DSNAME(OPTZOS.CLIST) DDNAME(SYSPRINT) NEW KEEP DSORG(PS)
           RECFM(F,B) LRECL(80)
     READY
     FTSHWOPT OUTPUT=*STDOUT(LAYOUT=*ZOS-PROC)
     READY
     FREE DDNAME(SYSPRINT)
     ```

   - Use FTSHWPTN to back up the partner list entries.
     These are the entries in the data set <openft qualifier>.<inst>.SYSPTF.

Example:

The output from the FTSHWPTN command is converted to the correct format using LAYOUT=*ZOS-PROC and redirected 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 you are using FTAC: Use FTEXPENV to back up the FTAC environment.
  These are the entries in the data set <openft qualifier>.<inst>.SYSFSA.

  ```
  FTEXPENV FTAC.SAVE
  ```

- Optionally: Use FTSHWLOG to back up the log file.
  These are the entries in the data set <openft qualifier>.<inst>.SYSLOG.

  ```
  FTSHWLOG SELECT=*ALL,NUMBER=*ALL,INF=*ALL,OUTPUT=*STDOUT(*CSV)
  ```

- Use the tools available in z/OS to back up the FT parameter library together with all its elements
  (data set <openft qualifier>.<inst>.PARM)

5. Use the TSO command FJGENPAR to output the installation parameters of the previous environment.
6. Fully shut down openFT:
   - Use the FTTERM command to terminate the started openFT job

   or

   - Terminate the started openFT task.
7. Uninstall openFT:
   - If the new version is to be installed under the same ID and with the same instance name as the old version then delete the openFT libraries and files belonging to the old version.

Changing to version 12.1

1. Install openFT version 12.1 and all the required components (see section “Installation of openFT” on page 37).
2. Use the FJGEN procedure to set up a new openFT instance and set the openFT installation parameters (see section “Setting openFT installation parameters with FJGEN” on page 56).
3. Adapt the FT parameter library:

   FJGEN recreates the FT parameter library. The library elements that are still required must be taken over from the old version. For a description of the structure of the FT parameter library and a detailed description of the library elements, see section “Setting up the FT parameter library” on page 57.

   Proceed as follows:

   - Prepare the OPFT subsystem (see page 92).
   - Set up openFT as a job or started task (see page 92).
   - Load and start the openFT load module (see page 96).
4. Import the following configuration data from the old version:
   - Operating parameter settings using the TSO command EXEC:

     ```
     EXEC OPTZOS
     ```

   - Partner list entries using the TSO command EXEC:

     ```
     EXEC PARTZOS
     ```

   - FTAC environment via FTIMPENV:

     ```
     FTIMPENV FTAC.SAVE
     ```
5. Start openFT (see page 96).

2.2 Generating the data communication system

Generating the data communication system for openFT comprises the following items:

- Extending the LOGON mode table and, if applicable, the COS table. This is only required if an SNA network is used for local communication or for communicating with partners.
- Generations for internal communication. Internal communication between the user commands and actual openFT processing can be performed either via VTAM or via TCP/IP (= default setting after installation) as required. To define which of these is to be used, you use the CMD_TRANS parameter in the PARM member of the openFT parameter library (see page 68). Generations are only necessary for VTAM; these are described in the following section
- Generation tasks for interconnection with partner systems via an SNA network; details are described in section “openFT interconnection via an SNA network” on page 31
- Generation tasks for interconnection with partner systems via a TCP/IP network; details are described in section “Interconnection via a TCP/IP network” on page 29

2.2.1 Extending the LOGON mode table (and, if applicable, the COS table)

The LOGON mode table of the VTAM generation must be extended by the following entry:

```
modtab   MODETAB
         MODEENT LOGMODE=FJMLMOD,                                  X
               FMPROF=X'03',                                       X
               TSPROF=X'03',                                       X
               PRIPROT=X'30',                                      X
               SECPROT=X'30',                                      X
               PSNDPAC=X'03',                                      X
               SRCVPAC=X'03',                                      X
               SSNDPAC=X'03',                                      X
               RUSIZES=X'....',                                    X
               COS=FTCOS
         MODEEND
```

The LOGMODE=FJMLMOD entry is essential.

The two macros MODETAB and MODEEND can be omitted if the MODEENT macro is inserted in an existing LOGON mode table.

The maximum lengths of the "request units" for the primary LU (first two bytes) and the secondary LU (last two bytes) specified in the RUSIZES parameter can assume values of between 1024 (X'87' for 8×2^7) and 32767 (e.g. X'FB' for 15×2^{11}). For further details, please refer to the IBM user guide for ACF/VTAM.

The specification of a "class of service table" (COS) for openFT is optional:

```
ISTSDCOS COSTAB
FTCOS    COS   VR=...
         COSEND
```

The two macros COSTAB and COSEND can be omitted if the COS macro is inserted in an existing "class of service table". For the virtual routes specified here, explicit routes must also be generated using the PATH macro.

Note that these entries - like all statements used for VTAM generation - must be entered in the correct column:

- statement: starting in column 10
- continuation lines: starting in column 16

2.2.2 Generations for internal communication

Only if internal openFT communication is performed via VTAM is it necessary to generate the VTAM applications for internal openFT data communication.

VTAM applications for internal openFT data communications

If internal openFT data communication is performed via VTAM then VTAM applications with the following predefined application names must be generated:

```
FJNADM (only for the STD instance)
FJNNDMS0
FJNNDMS1
 .
 .
 .
FJNNDMS9
FJAftid
FJDftid
```

Where ftid is the FT identifier. This alphanumeric character string may consist of a maximum of five characters and must be unique for all linked FT systems. This ftid must also be specified in the FJGEN command (see page 104) for the openFT instance. The entries FJAftid and FJDftid must exist for each openFT instance that uses internal communication via VTAM. These are the instances for which CMD_TRANS=TCP is **not** set in the PARM member of the parameter library.

The VTAM applications FJNADM, FJNNDMS0,... , FJAftid and FJDftid are used for internal openFT communication. FJNADM is only used by the instance STD. As a minimum requirement, you must generate the applications FJNADM and FJNNDMS0. Up to 10 applications (in continuous ascending order from FJNNDMS0 to FJNNDMS9) can be generated. This provides users with more connections for the entry of the commands (this also applies to the program interface and menu interface).

These VTAM applications are generated using the following statements:

```
         VBUILD   TYPE=APPL
FJNADM   APPL     ACBNAME=FJNADM,                          X
                  AUTH=(ACQ,VPACE),                        X
                  DLOGMOD=FJMLMOD,                         X
                  MODETAB=modtab,                          X
                  PRTCT=ft-password,                       X
                  VPACING=3
FJNNDMS0 APPL     ACBNAME=FJNNDMS0,                        X
                  AUTH=(ACQ,VPACE),                        X
                  DLOGMOD=FJMLMOD,                         X
                  MODETAB=modtab,                          X
                  VPACING=3
```

```
FJNNDMS1 APPL      ACBNAME=FJNNDMS1,                          X
                   AUTH=(ACQ,VPACE),                          X
                   DLOGMOD=FJMLMOD,                           X
                   MODETAB=modtab,                            X
                   VPACING=3
FJNNDMS2 APPL      ACBNAME=FJNNDMS2,                          X
                   AUTH=(ACQ,VPACE),                          X
                   DLOGMOD=FJMLMOD,                           X
                   MODETAB=modtab,                            X
                   VPACING=3
.
.
.
FJAftid  APPL      ACBNAME=FJAftid,                           X
                   AUTH=(ACQ,VPACE),                          X
                   DLOGMOD=FJMLMOD,                           X
                   MODETAB=modtab,                            X
                   PRTCT=ft-password,                         X
                   VPACING=3
FJDftid  APPL      ACBNAME=FJDftid,                           X
                   AUTH=(ACQ,VPACE),                          X
                   DLOGMOD=FJMLMOD,                           X
                   MODETAB=modtab,                            X
                   PRTCT=ft-password,                         X
                   VPACING=3
```

where

modtab
: is the name of the LOGON mode table (see page 26),

ft-password
: is the FT password which can be used to protect the VTAM applications, the request file, the partner list and the trace files of an openFT installation. The password is specified in the installation parameters (see page 57) or in the FT administration command FJGEN (see page 104). The applications FJNNDMS0 ... FJNNDMS9 must not be protected by an FT password. For all other applications, password protection is optional.

The entry DLOGMOD=FJMLMOD is essential in these statements.

2.2.3 Interconnection via a TCP/IP network

The following generation operations must be performed in order to connect to FT partner systems via a TCP/IP network:

– The connection between openFT (z/OS) and the software product TCP/IP for MVS must be generated.

– The transport system address of the local openFT- instance must be determined.

– The transport system addresses of the remote partner systems must be entered.

2.2.3.1 Transport system address of the local openFT instance

In the case of a TCP/IP interconnection, the transport system address of a local openFT instance consists of the Internet address, port number and T-selector.

The **Internet address** of the z/OS system on which the local openFT instance is running is assigned using the HOST NAME parameter in the FJGEN command (see page 104). In HOST NAME, you should always directly specify the IP address or the host name. Multiple parallelly running openFT instances must have different IP addresses. Please note that you may only use IP addresses that are defined in your z/OS system's address space.

The **port number** of the openFT (z/OS) main station (passive port) is defined using the OPENFT-APPL parameter in the FTMODOPT command. We strongly recommend to use the default port number 1100. This is predefined as default in all openFT products and therefore greatly simplifies addressing in a heterogeneous environment.

The **T-selector** of the openFT (z/OS) main station has the name $FJAM␣␣␣.

2.2.3.2 Transport system addresses for TCP/IP partner systems

The transport system address of a TCP/IP partner system consists of the Internet address of the remote computer, the T-selector of the remote FT system and, where appropriate, the port number of the RFC1006 implementation of the remote FT system if this differs from the default port number 1100. The transport system addresses of all partner systems which are to be accessed via TCP/IP must be reported to openFT (z/OS). This has been considerably simplified as of openFT V10:

- The FT administrator enters the partner systems in the partner list with the FTADDPTN command and stores the necessary address information there. For further details on specifying addresses, see page 130.

  If host names are used, it must be possible to determine the associated IP address from the relevant data source, e.g. from the file TCPIP.HOSTS.LOCAL or using DNS.

- If dynamic partners are permitted (see FTMODOPT command), it is also possible to directly address a partner system of this type without it being entered in the partner list.

2.2.4 openFT interconnection via an SNA network

Further VTAM applications must be generated for interconnection with FT partner systems via an SNA network, irrespective of the type of partner system in question (openFT (z/OS), openFT (Windows) and the connection method of Microsoft's Host Integration Server used there):

– a main station that receives all external transfer requests ("inbound submissions")

– substations that are used for transfer requests initiated by the local openFT instance ("outbound submissions").

Naming conventions

The names of these VTAM applications are formed in accordance with a set of naming conventions. They start with a prefix (main station: FJM..., substations: A01..., A02... etc.) to which the ftid of the local openFT instance is added (see section "Generations for internal communication" on page 27).

This convention makes it possible to assign unique names to VTAM applications. These names are then used by the local openFT instance when establishing the SNA connection to remote systems.

It also provides unique identification of all interconnected partner systems at transport system level and therefore facilitates extended authentication (see page 136). If extended authentication is enabled in a remote FT system then, in the case of an SNA connection, the remote system will only accept inbound requests if it is able to assign the substation name (A01ftid, A02ftid etc.) that appears as the sender address to the name of the partner system's main station entered in the partner list (FJMftid). You can activate extended authentication in z/OS by using PARTNER-CHECK=*TRANSPORT-ADDRESS in the command FTMODOPT. It is only possible to deviate from these main station and substation names ("free VTAM names"), which apply throughout the SNA network, if extended authentication is not used. For further information, refer to the notes below.

The following VTAM applications must therefore be generated:

```
FJMftid
A01ftid
A02ftid
  .
  .
  .
Annftid
```

where:

ftid
: FT identifier. This alphanumeric character string can consist of up to five characters and must be unique among all interconnected FT systems.

nn
: is greater than or equal to the maximum number of transport connections (defined with the operating parameter CONNECTION-LIMIT, see FTMODOPT command). These numbers (01, 02,... nn) must be assigned in continuous ascending order as otherwise not all the generated VTAM applications can be used by openFT.

These VTAM applications are generated using the following statements:

```
FJMftid  APPL      ACBNAME=FJMftid,                          X
                   AUTH=(ACQ,VPACE),                         X
                   DLOGMOD=FJMLMOD,                          X
                   MODETAB=modtab,                           X
                   PARSESS=YES                               X
                   PRTCT=ft-password,                        X
                   VPACING=3
A01ftid  APPL      ACBNAME=A01ftid,                          X
                   AUTH=(ACQ,VPACE),                         X
                   DLOGMOD=FJMLMOD,                          X
                   MODETAB=modtab,                           X
                   PARSESS=YES                               X
                   PRTCT=ft-password,                        X
                   VPACING=3
A02ftid  APPL      ACBNAME=A02ftid,                          X
                   AUTH=(ACQ,VPACE),                         X
                   DLOGMOD=FJMLMOD,                          X
                   MODETAB=modtab,                           X
                   PRTCT=ft-password,                        X
                   VPACING=3
.
.
.
Annftid  APPL      ACBNAME=Annftid,                          X
                   AUTH=(ACQ,VPACE),                         X
                   DLOGMOD=FJMLMOD,                          X
                   MODETAB=modtab,                           X
                   PRTCT=ft-password,                        X
                   VPACING=3
```

where:

modtab
: is the name of the LOGON mode table (see page 26),

ft-password
: is the FT password which may be used to protect the VTAM applications, the request file, the partner list and the trace files of an openFT installation. The password is specified in the FT parameter library (see page 57) or in the FT administration command FJGEN (see page 104). This password specification is optional for the VTAM applications described here.

The entry DLOGMOD=FJMLMOD is essential in these statements.

Notes

- During VTAM or NCP generation, it is also necessary to enter the main station and substations of all FT partner systems which are connected via SNA. However, these entries depend on the type of partner system in question (openFT (z/OS), openFT (Windows)) and on the connection method used (Host Integration Server from Microsoft). These entries are therefore described in the sections dealing with the individual partner systems (page 35).
- Entries in partner systems using openFT V10 and higher: The name of the main station (FJMftid) of the local openFT instance can be specified in the partner list of the remote FT system (e.g. in the PARTNER-ADDRESS operand of the FTADDPTN command). Examples of the interrelation of VTAM generation and the partner list entries can also be found in the sections dealing with the individual partner systems (page 35).
- Entries in partner systems using openFT < V10: The name of the main station (FJMftid) of the local openFT instance must also be specified in the network description of the remote FT system, e.g. in the NETWORK-ADDRESS operand of the FTADDPTN command.
- You are advised to generate at least 8 applications A01..., A02..., A03... etc. in order to avoid bottlenecks. The substations of the local openFT instance, A01ftid to Annftid, must be numbered in continuous ascending order, otherwise not all the applications and transport connections available can be used.
- In general, the names formed in accordance with the naming conventions described above must be entered both as the name of the VTAM application and as the value of the ACBNAME name (see APPL statements above). Only then can the remote FT system perform extended authentication (see page 136) as specified in case of a remote openFT (z/OS) in the PARTNER-CHECK=*TRANSPORT-ADDRESS parameter of the FTMODPTN command.

However, if "free" main station and substation names (which apply throughout the SNA network) are to be used, then the following applies:

– The "free name" is specified as the name in the APPL statement. However, the name formed in accordance with the naming conventions described above must still be specified for the ACBNAME parameter.

– The remote FT systems must/can contain corresponding entries in the network description file/partner list. The entry is mandatory in systems using openFT < V10 and is optional in systems using openFT V10 and higher. If the remote FT system is also an openFT (z/OS), the "free VTAM name" of the local openFT instance's main station must/can also be specified in the FTADDPTN command (as a value for the NETWORK-ADDRESS parameter *in* openFT < V10 or, as of V10, as a value of the PARTNER-ADDRESS parameter). If openFT (z/OS) V6 or earlier is used in the remote system then the "free VTAM name" of the local system's main station must be specified there as a value for the RELADR parameter in the FJADDSYS command. The name of the local system's main station formed in accordance with the naming conventions must still be specified as SYSADR, i.e. FJMftid.

– If a connection is made to openFT (z/OS) or to openFT V8.1 for Windows with Microsoft's Host Integration Server, then the corresponding entries in the conversion tables or name servers of these products must be made.

– If a "VTAM Interpret Table" is generated in the local system, it must also contain the names of the openFT VTAM applications

```
FJMftid
A01ftid
A02ftid
etc.
```

– If "free VTAM names" are used for the stations of the local openFT instance, none of the partner systems to which this system is connected may operate extended authentication.

The following sections deal with the individual partner systems and contain examples of "free VTAM names".

"Free VTAM names" can only be used for main stations and substations (FJMftid, Annftid). They cannot be used for VTAM applications for internal openFT data communications (FJNADM, FJNNDMS0 ... FJNNDMS9, FJAftid, FJDftid)

2.2.4.1 openFT interconnection of two z/OS systems via an SNA network

No extensions to the VTAM generation beyond the entries necessary for the homogeneous interconnection of two z/OS systems and the extensions to the LOGON mode table and the VTAM applications described above are required when the two openFT (z/OS) systems are to be interconnected via an SNA network.

Since the interconnected systems are located in different domains, the VTAM applications used for data transfer (FJMftid, A01ftid, A02ftid, etc.) must be defined as "cross domain resources".

Example

Linking two systems openFT (z/OS):
FTZOS1 (ftid = *ZOS1*) and *FTZOS2* (ftid = *ZOS2*).

– The VTAM applications FJNADM, FJNNDMS0 (at least), FJAZOS1, FJDZOS1, FJMZOS1, A01ZOS1, ... , A08ZOS1 must be generated in VTAM on the z/OS computer with the FT system *FTZOS1* (specify PARSESS=YES with FJMZOS1, A01ZOS1).

– The VTAM applications FJNADM, FJNNDMS0 (at least), FJAZOS2, FJDZOS2, FJMZOS2, A01ZOS2, ... , A08ZOS must be generated in VTAM on the z/OS computer with the FT system *FTZOS2* (specify PARSESS=YES with FJMZOS2, A01ZOS2).

– The VTAM applications used for transport (FJMZOS1, A01ZOS1, ..., A08ZOS1, FJMZOS2, A01ZOS2, ... , A08ZOS2) must additionally be defined as "cross domain resources".

– In addition, the LOGON mode table for openFT must be generated in both computers.

– The remote FT system *FTZOS2* can be entered as follows in the partner list of the FT system *FTZOS1*:

```
FTADDPTN PARTNER-NAME=FTZOS2,PARTNER-ADDRESS=FJMZOS2:SNA
,IDENTIFICATION= ...
```

If you address FTZOS2 directly in FT requests then specify FJMZOS2:SNA.

– The remote FT system *FTZOS1* can be entered as follows in the partner list of the FT system *FTZOS2*:

```
FTADDPTN PARTNER-NAME=FTZOS1,PARTNER-ADDRESS=FJMZOS1:SNA
,IDENTIFICATION= ...
```

If you address FTZOS1 directly in FT requests then specify FJMZOS1:SNA.

– In each case, the instance ID of the partner system in the IDENTIFICATION parameter of the FTADDPTN command must be entered in the same way as it has been defined there in the FTMODOPT command.

The same example using "free VTAM names"

The following names, which apply throughout the network, should be used for the *FTZOS1* FT system:

MVSMAIN	(for FJMZOS1)
MVSSUB1	(for A01ZOS1)
MVSSUB2	(for A02ZOS1)
...	...
MVSSUB8	(for A08ZOS1)

The VTAM-APPL statements for these applications on the computer with the FT system *FTZOS1* are as follows:

```
MVSMAIN  APPL   ACBNAME=FJMZOS1, ...
MVSSUB1  APPL   ACBNAME=A01ZOS1, ...
MVSSUB2  APPL   ACBNAME=A02ZOS1, ...
...
MVSSUB8  APPL   ACBNAME=A08ZOS1, ...
```

The remote openFT system *FTZOS1* can be entered as follows in the partner list of the openFT system *FTZOS2*:

```
FTADDPTN PARTNER-NAME=FTZOS1,PARTNER-ADDRESS=MVSMAIN:SNA
,IDENTIFICATION='ZOS1.FUSINET.AT'
```

2.3 Installation of openFT

2.3.1 Preparations for installation

Before installing the product, you should make a few preparations, such as defining the admissions for the openFT user IDs and the openFT privileges and regulating protection of the administration files.

Migrating from an older version

Update installations are not supported. Back up the configuration data, for example the operating parameters, partner list entries and, if applicable, the FTAC environment as described in section “Change of version” on page 22.

2.3.1.1 User IDs for openFT

The following user IDs are required for openFT operation:

– a user ID under which openFT runs (as a job or started task, see page 92)

– one or more FT administrator IDs

– one or more FTAC administrator IDs (only if the FTAC functionality is used)

openFT can run under an ID without TSO authorization, if this is required on account of the privilege level which this user ID needs (see next section).

The user IDs that openFT or FTAC can administer are defined in the FTADM and FTACADM members of the FT parameter library PARM (see page 70).

The internal data sets that are required to operate the openFT instances are catalogued with the prefix OPENFT QUALIFIER. In particular, this applies to the instance’s request file and the partner list (see section "Internal openFT data sets" on page 193). The OPENFT QUALIFIER is specified in the FJGEN command (see page 56 and page 104).

2.3.1.2 openFT privileges

When installing openFT it is important to note the following points concerning privileges:

- If the product RACF (or compatible product) is installed in the system, the OPENFT load module must be stored in a library which is subject to APF authorization, since it accesses privileged RACF macros (see the section “Linking openFT with data protection products” on page 97). In addition, the OPENFT load module must possess the linkage editor attribute "AC(1)". The OPENFT load module supplied already has this attribute.

  openFT must also have APF authorization in order to perform the following functions:

  – transfer a complete PO or PDSE data set
  – charge file transfer requests (write account records to the SMF file)
  – output asynchronous messages after termination of a transfer request to the TSO user whose user ID was specified in the TRANSFER-ADMISSION of the system involved and/or to one or several consoles.

  In addition to the library containing the OPENFT load module, the other libraries of the library hierarchy STEPLIB, TASKLIB, JOBLIB ... APF must also be authorized, i.e.:

  – the library containing openFT as a subsystem, known as the LPALIB
  – the library containing the OPENFTCR load module (see section “Installation of the openFT-CR delivery unit” on page 48)

- Since openFT uses socket calls to establish TCP/IP connections, the user ID under which openFT runs (as a job or as a started task, see page 92) also needs an OMVS segment (OMVS: OpenEdition MVS). No special privileges are needed, i.e. any UID (OMVS user ID) can be used. The user ID must belong to a group for which a GID (OMVS group ID) has been defined. The GID is defined with RACF; see also IBM manual "OpenEdition Planning", chapter "Controlling OpenEdition Security".
- If the file SYS1.UADS is installed in the system and is to be used by openFT, the user ID under which openFT is running (as a job or started task, see page 92) must be granted read access to this file.
- In an z/OS system with RACF (Resource Access Control Facility), the user ID under which openFT is running must also be authorized to access the files and volumes of all openFT users if these are protected by RACF. In particular it must be granted:

  – read access (READ) to send files
  – write access (ALTER) to receive files

  The z/OS administrator can assign specific access rights to these files and to the associated data volumes. However, it is considerably easier to assign the RACF attribute OPERATIONS to the user ID under which openFT is running. If this approach is taken, it is advisable to not to assign any TSO authorization to this user ID for reasons

of data security. Even if the user ID under which openFT is running possesses the RACF OPERATIONS attribute and is therefore able to access all the files in the system, there is no danger of FT user transfer requests infringing on data security, since openFT verifies the validity of all the data access attempts that occur during file transfer (see section “Linking openFT with data protection products” on page 97).

The same rules apply to products compatible with RACF. For further information please refer to the product-specific manuals.

2.3.1.3 Protecting openFT administrative files

The data sets created for the administration and operation of openFT should be protected against unauthorized access (e.g. by using RACF). The degree of protection needed will vary depending on the particular security requirements of individual computer centers. The following sections contain recommendations for protecting the most important data sets. For some of the data sets, the most stringent access restrictions that will still allow openFT operation are described.

FT parameter library

The parameters with which openFT is adjusted to installation-specific requirements (see section “Setting up the FT parameter library” on page 57) are stored in the FT parameter library. This is highly sensitive information, the integrity of which is absolutely essential for openFT to be able to function properly (for instance the list of FT or FTAC administrators and possibly the name the FTAC file; see below). This file must therefore be protected extremely carefully.

Request file, partner list, operational parameters file

The request queue, the partner list and the operational parameters file are three DA data sets set up automatically under the following names the first time the system is started:

– The request queue '<openft qualifier>.<inst>.*SYSRQF*'

– The partner list '<openft qualifier>.<inst>.*SYSPTF*'
The partner list contains the address information for the partner systems and corresponds to the network description file used in previous openFT versions.

– The operational parameters file '<openft qualifier>.<inst>.*SYSOPF*'.

Here, <openft qualifier> is the prefix with which the openFT administrative files are created (OPENFT QUALIFIER in the FJGEN command). <inst> is the instance name (INSTANCE NAME in the FJGEN command).

These three files only need to be accessed by the user ID under which openFT is running.

Logging file

The logging file is generated automatically by openFT. Its components are described in section “Internal openFT data sets” on page 193.

Usually, the names of the components of the logging files all begin with ’<openft qualifier>.<inst>.SYSLOG'. "openft qualifier" is the prefix with which the openFT administrative files are created (OPENFT QUALIFIER in the FJGEN command). "inst" is the instance name (INSTANCE NAME in the FJGEN command). Instead of the usual second level qualifier inst.SYSLOG, the administrator may allocate a different name to the file (LOGFILE_2ND_Q key in the PARM member of the FT parameter library).

Only the user ID under which openFT is running should be able to access the components of the logging files. Please also read the note at the end of section "FTAC files".

If you want to store the logging records permanently, redirect the output from the FTSHWLOG command to a file and then back up this file or use the new logging functionality introduced in Version 12 to back up logging records in offline logging files.

To prevent the logging file from becoming unnecessarily large, you should occasionally use the FTDELLOG command to delete old logging records or change the logging file from time to time using the command FTMODOPT LOGGING=*CHANGE-FILES (see page 141) and archive logging files that are no longer online as required.

FTAC file

The FTAC file is generated automatically by openFT when FTAC is used. It contains the FTAC environment, i.e. the admission sets, admission profiles, etc. The components of the file are described in section “Internal openFT data sets” on page 193.

The names of the components of the FTAC file all begin with '<openft qualifier>.<inst>.SYSFSA'. "openft qualifier" is the prefix with which the openFT administrative files are created (OPENFT QUALIFIER in the FJGEN command). "inst" is the instance name (INSTANCE NAME in the FJGEN command). Instead of the usual second level qualifier inst.SYSFSA, the administrator may allocate a different name to the file (FILE_2ND_Q key in the FTACPAR member of the FT parameter library).

For reasons of security it is strongly recommended that the components of this file be accessible only to the main FT administrator ID and the user ID under which openFT runs.

Note

If you are using RACF and you want to protect the logging file and the FTAC file using generic profiles, you must make sure that all components of the files are covered by the names of the generic profiles.

If you want to use to implement a standard protection for the request file, the partner list, the logging file and the FTAC file and if you select the same beginning for the file names of all of these files then you will need only two generic profiles to protect them.

If you use the standard file names for the files, you only need to implement the following generic profiles for the individual openFT instances:

'<openft qualifier>.<inst>.SYS*'

This generic profile protects the request file (SYSFSF), the partner list (SYSPTF) and the PS data sets that are part of the logging file and the FTAC file (SYSLOG and SYSFSA).

'<openft qualifier>.<inst>.SYS*.*'

This generic profile protects the components of the VSAM cluster, which are part of the logging file and the FTAC file (SYSLOG.P00 etc. for the logging file, SYSFSA.P00 etc. for the FTAC file).

The OPENFT QUALIFIER stands for the file name prefix defined in the FJGEN command, while inst refers to the instance name defined for the corresponding openFT instance in the INSTANCE NAME parameter in the FJGEN command.

2.3.2 Installing from CD

openFT (z/OS) is supplied for installation with SMP/E (System Modification Program/Extended) as a "Custom-Built Product Delivery Offering" (CBPDO) as "function SYSMOD" with the following characteristics:

File name prefix (RFDSNPFX):	OPENFT
Identification (FMID):	OFT121A

In order to install openFT, an SMP/E environment for openFT has to be created; amongst other things, this comprises a product-specific "Consolidated Software Inventory" (CSI). A set of procedures is supplied with which an SMP/E environment for openFT is created and with which the SMP/E statements RECEIVE, APPLY and ACCEPT are executed.

openFT (z/OS) including all the additional delivery units is supplied as standard only on CD. Therefore, you must copy the product files from the CD on a Unix or Windows computer and then transfer these to the z/OS computer and unpack them.

2.3.2.1 Transferring files from CD to the z/OS computer and unpacking them

Insert the product CD in a Windows or Unix system and proceed as follows:

1. Copy the files from the CD to the Windows or Unix system.
2. Transfer all the files to the z/OS computer on which you want to install openFT. This also includes the procedures which are located in the TOOLS directory on the CD and are needed for the unpacking and loading operation, see step 3.

   To perform the transfer, you can use, for example, the openFT version on the relevant system or the transfer function provided by a 3270 emulation. Please note that the XMIT files always have to be transferred on binary format and the CLIST files in text format. If you use openFT, specify the options -b (binary) and -r=f80 (fixed record length 80) in the transfer command for the XMIT files.

   Examples

   You want to transfer an XMIT file with openFT (Windows) to z/OS using the *ncopy* or *ft* command:

   ```
   ncopy OFT121A.F1.XMIT zospart!% uid,,passwd -b -r=f80

   ft OFT121A.F1.XMIT zospart!OFT121A.F1.XMIT uid,,passwd -b -r=f80
   ```

   If you use the openFT Explorer then you should enter the following in the transfer request:
   File Type: **Binary** (General tab)
   Maximum Record Length: **f80** (Options tab)

3. Execute the procedure FTLOAD.CLIST on the z/OS computer:

   ```
   EXEC FTLOAD
   ```

   This procedure unpacks and loads the XMIT files. This procedure is transferred during step 2 and is located in the TOOLS directory on the CD.

   

   There are separate procedures for unpacking and loading each of the other components: openFT-CR, openFT-AC and openFT-FTP.

2.3.2.2 openFT product files

After unpacking, the following product files are available for openFT:

OPENFT.OFT121A.SMPMCS
MCS statements for SMP/E (MCS: Modification Control Statement)

OPENFT.OFT121A.F1
PO data set containing the following CLIST's:

OFT121A	JCLIN statements for transferring the other product modules from the tape with SMP/E (JCLIN: Job Control Input)
OPFT#01	creates the system and backup files for openFT
OPFT#02	installs the SMP/E environment for openFT
OPFT#03	initializes the SMP/E environment for openFT
OPFT#04	executes the SMP/E statement RECEIVE
OPFT#05	executes the SMP/E statement APPLY
OPFT#06	executes the SMP/E statement ACCEPT

OPENFT.OFT121A.F2
PO data set containing the following members:

FGMD	contains a Web link to the Release Notice in German
FGME	contains a Web link to the Release Notice in English

OPENFT.OFT121A.F3
PO data set containing samples for FT users and FT administrators (the $$INDEX member contains a brief description of the other members).

OPENFT.OFT121A.F4
FT basic procedure library with the CLIST FJGEN. The other CLIST procedures are stored in instance-specific FT procedure libraries during the FJGEN run.

OPENFT.OFT121A.F5
openFT Load module library containing the following members:

OPENFT	Program openFT **without** SSL encryption
OPENFTS	Alias name for OPENFT (see Explanation (1) below)
OPENFTSL	Program openFT **with** SSL encryption
OPENFTSS	Alias name for OPENFTSL (see Explanation (1) below)
OPFTSUBL	Subsystem handler

(1) The aliases OPENFTS and OPENFTSS are identical to the entries OPENFT and OPENFTSL in terms of their functions. The alias can be used in place of this entry if it is necessary to refer to the load module using a name other than the user ID under which the openFT job is running.

OPENFT can also be used for console applications or NetView applications.

OPENFT.OFT121A.F6
Load module library containing the members
FTADDPTN, FTADM, FTMODREQ, FTMODOPT, FTMODPTN, FTREMPTN, FTSHWMON, FTSHWOPT, FTSHWPTN, FTSHWD, FTSTART, FTSTOP, FTCREKEY, FTDELKEY, FTUPDKEY, FTSHWLOG, FTSHWNET, FTSHWINS, FTUPDPAR, FTDELLOG, FTHELP, FTTERM, NCOPY, NSTATUS, NSTAT, NCANCEL, NCAN, FTSCOPY, FTACOPY, FTCANREQ, FTSHWREQ, FTDEL, FTMOD, FTSHW, FTCREDIR, FTMODDIR, FTDELDIR, FTUPDKEY, FTEXEC, FTTRACE, OPFTSUBL, FTMODKEY, FTSHWKEY, FTIMPKEY.
The following alias names can only be used if openFT-AC is installed
FTCREPRF, FTDELPRF, FTMODADS, FTMODPRF, FTSHWADS, FTSHWPRF, FTSHWRGE, FTEXPENV, FTIMPENV, FTSHWENV.

OPENFT.OFT121A.F7
Macro library containing the ASSEMBLER macro OPENFT.

OPENFT.OFT121A.F8
Library containing the ISPF panel definitions for the menu interface for FT users and FT administrators

OPENFT.OFT121A.F9
Library containing the corresponding CLIST procedures.

OPENFT.OFT121A.F10
Library containing the corresponding message definitions.

OPENFT.OFT121A.F11
Contains the members IGX00211, OPFTIGX, OPFTINIT and OPFTSUB. These objects are installed in a separate PO library with the name LPALIB during installation. Users can then copy them to any location (e.g. to SYS1.OPENFT.LPALIB) in order to then start openFT as a subsystem.

OPENFT.OFT121A.F12
Contains the code tables IBM037, IBM273 and IBM500.

OPENFT.OFT121A.F13
Contains the TCL scripts RFC1006, NEABF, FTCMD and FTPING, see also note on page 45.

2.3.2.3 How to proceed with the installation

Proceed as follows to install openFT:

The procedure OPFTTEMP.OFT121A.F1 can be used to perform the installation.

The following steps are necessary in order to install openFT:

1. If an openFT version is already present, delete it together with all the additional delivery units.
2. Specify the required installation directory or installation prefix by setting the T_BASE variable in the procedure OPFTTEMP.OFT121A.F1.

   The default settings are the user ID and directory in which the temporary installation files are located.
3. Call the following procedures to install openFT:

```
EXEC 'USERID.OPFTTEMP.OFT121A.F1(OPFT#01)'
EXEC 'USERID.OPFTTEMP.OFT121A.F1(OPFT#02)'
EXEC 'USERID.OPFTTEMP.OFT121A.F1(OPFT#03)'
EXEC 'USERID.OPFTTEMP.OFT121A.F1(OPFT#04)'
EXEC 'USERID.OPFTTEMP.OFT121A.F1(OPFT#05)'
EXEC 'USERID.OPFTTEMP.OFT121A.F1(OPFT#06)'
```

If the optional delivery units openFT-CR (see page 48), openFT-AC (see page 51) or openFT-FTP (see page 53) are not to be installed, you can now continue with the steps described in section “Making the commands and the ISPF panels available” on page 46, and section “Setting openFT installation parameters with FJGEN” on page 56 etc.

Note

- If openFT is to run with APF-authorization, then the following libraries must have APF authorization:
  - the library that contains the OPENFT or OPENFTSL load module (i.e. the library to which the OPENFT.OFT121A.F5 library was copied from CD)
  - and the library that contains openFT as a subsystem (i.e. the library to which the library OPENFT.OFT121A.F11 is copied from CD).

  This also applies to the other libraries in the library hierarchy STEPLIB, TASKLIB, JOBLIB ... (see section “openFT privileges” on page 38).
- If FTPING is to be used, you have to install the TCL utility in z/OS OMVS first. The TCL utility can be downloaded from Rocket Software link http://www.rocketsoftware.com/ported-tools/tcl-8511.

2.3.3 Making the commands and the ISPF panels available

When the openFT product has been read in, you must make sure that the library containing the openFT load modules and the openFT commands (OPENFT.LOAD and OPENFT.NCLOAD) and the libraries containing the ISPF panels, CLISTs and messages for the FT administrator menu interface are available to the user IDs which are authorized to use these commands or this menu interface. In other words, you must concatenate the libraries accordingly.

2.3.3.1 Concatenating libraries with the openFT commands

This section contains a description of two different ways of making the CLIST libraries containing the administration commands (FT procedure library) available to those user IDs that are authorized to use these commands.

Modifying the LOGON procedure

The LOGON procedure is executed each time a user logs on. During this procedure the FT procedure library is concatenated with the library containing the defined system procedures ('system-procedure-library'):

```
//SYSPROC  DD  DSN=system-procedure-library,DISP=SHR
//         DD  DSN=ft-procedure-library,DISP=SHR
//OPENFT   DD  DSN=<openft qualifier>.<inst>.CONN,DISP=SHR
```

If you want to use the menu interface for the FT administrator, you must make some further modifications to the LOGON procedure (see section “Concatenating libraries with the openFT commands” on page 46**).**

Making the library available dynamically in a TSO session

The library can alternatively be made available during a TSO session by means of the following TSO commands:

```
ALLOC FILE(SYSPROC) DSNAME('system-procedure-library' -
'ft-procedure-library') REUSE SHR
```

and the two commands:

```
ALLOC DSNAME('<openft qualifier>.<inst>.CONN') DDNAME(OPENFT) SHR REUSE
```

```
TSOLIB ACT DATASET('<openft qualifier>.OPENFT.NCLOAD')
```

Following this, the FT commands are available during the current session.

You are advised to protect the library containing these CLISTs, e.g. by means of RACF, in order to protect these procedures and thus the entire FT administration against unauthorized access.

2.3.3.2 Concatenating libraries containing ISPF panels

The ISPF panel definitions and the associated MSG and CLIST libraries are contained in three PO data sets:

```
OPENFT.OFT121A.F8  contains the panel definitions
OPENFT.OFT121A.F9  contains the CLIST procedures required for execution
OPENFT.OFT121A.F10  contains the message definitions
```

These libraries, into which the components were read from the openFT CD (see section “Installing from CD” on page 42), must be made accessible to those user IDs who are authorized to administer openFT via the menu interface, using the methods already described above. This description deals only with the modification of the LOGON procedure; the information given above on making the library available during a TSO session also applies here.

Modification of the LOGON procedure

```
//SYSPROC  DD  DSN=system-procedure-library,DISP=SHR
//         DD  DSN=ft-procedure-library,DISP=SHR
//         DD  DSN=ft-clist-library,DISP=SHR
//OPENFT   DD  DSN=<openft qualifier>.<inst>.CONN,DISP=SHR
//ISPPLIB  DD  DSN=system-panel-library,DISP=SHR
//         DD  DSN=ft-panel-library,DISP=SHR
//ISPMLIB  DD  DSN=system-message-library,DISP=SHR
//         DD  DSN=ft-message-library,DISP=SHR
```

The "ft-procedure-library" is the library described above for the FT administration commands. "ft-clist-library" is the above-mentioned library for the CLIST procedures in the openFT menu interface (after installation, OPENFT.PANEL.CLIST), "ft-panel-library" is the library for panel definitions (after installation, OPENFT.PANELS) and "ft-message-library" is the library for messages (after installation, OPENFT.PANEL.MSG). These libraries, too, should be protected against unauthorized access, e.g. using RACF.

CLIST command procedures for the FT administration commands are created when the FJGEN procedure is executed (see page 56) and stored in the library <openft qualifier>.<inst>.CLIST.

Starting the panel interface

Call the following command under TSO:

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

By calling this start CLIST, you can access the initial panel of the openFT menu interface both under TSO and via the general ISPF/PDF interface (enter "TSO EXEC clistname" in the command line). For further information, please refer to the relevant IBM manuals.

2.4 Installation of the openFT-CR delivery unit

If openFT is also to be able to transfer job data (file contents) in encrypted form in file transfer requests, the openFT-CR delivery unit must be installed.

The openFT-CR delivery unit is supplied on CD as a separate order.

The delivery unit openFT-CR is installed with SMP/E as a supplement to the function SYSMOD for openFT described in section “Installing from CD” on page 42. So, in order to be able to install the delivery unit openFT-CR, openFT must be available in an SMP/E environment.

openFT (z/OS) including all the additional delivery units is supplied as standard only on CD. Therefore, you must copy the product files from the CD on a Unix or Windows computer and then transfer these to the z/OS computer and unpack them.

Proceed as follows:

1. Perform steps 1 and 2 in section “Transferring files from CD to the z/OS computer and unpacking them” on page 42.
2. Call the procedure FTCRLOAD.CLIST:

   ```
   EXEC FTCRLOAD
   ```

   This procedure unpacks and loads the XMIT files. This procedure is transferred during step 1 and is located in the TOOLS directory on the CD.

2.4.1 openFT-CR product files

After unpacking, the following product files are available for openFT-CR:

OPENFT.OFC121A.SMPMCS
MCS statements for SMP/E

OPENFT.OFC121A.F1
PO data set containing the following CLIST's:

OFC121A	JCLIN statements for transferring the other product modules from the tape with SMP/E
OPFTCR#1	creates the system and backup files for openFT-CR
OPFTCR#2	extends the SMP/E environment for openFT by the entries required for openFT-CR
OPFTCR#3	executes the SMP/E statement RECEIVE
OPFTCR#4	executes the SMP/E statement APPLY
OPFTCR#5	executes the SMP/E statement ACCEPT

OPENFT.OFC121A.F2
PO data set containing the following members:
FGMD#CR contains a Web link to the Release Notice in German
FGME#CR contains a Web link to the Release Notice in English

OPENFT.OFC121A.F3
Load module library containing the following member:
OPENFTCR Load module for openFT-CR

2.4.2 How to proceed with the installation

Proceed as follows to install openFT-CR:

The procedure OPFTTEMP.OFC121A.F1 can be used to perform the installation.

The following steps are necessary in order to install openFT-CR:

1. Specify the required installation directory or installation prefix by setting the T_BASE variable in the procedure OPFTTEMP.OFC121A.F1.

   The default settings are the user ID and directory in which the temporary installation files are located.

2. Call the following procedures to install openFT-CR:

```
EXEC 'USERID.OPFTTEMP.OFC121A.F1(OPFTCR#1)'
EXEC 'USERID.OPFTTEMP.OFC121A.F1(OPFTCR#2)'
EXEC 'USERID.OPFTTEMP.OFC121A.F1(OPFTCR#3)'
EXEC 'USERID.OPFTTEMP.OFC121A.F1(OPFTCR#4)'
EXEC 'USERID.OPFTTEMP.OFC121A.F1(OPFTCR#5)'
```

3. Copy the load module OPENFTCR that is present in the library OPENFT.OFC121A.F3 to the library OPENFT.LOAD or to a library concatenated with this. To load this module, openFT calls the system macro LOAD, which searches for a member with the name OPENFTCR in the conventional library hierarchy STEPLIB, TASKLIB, JOBLIB ... . If openFT is to run with "APF authority", the library that contains the OPENFTCR load module must also be APF-authorized (see section “openFT privileges” on page 38).

   The OPENFTCR module can be added or removed when the local openFT instance is deactivated, i.e.:

   – openFT must be stopped using the FTSTOP command and
   – the started openFT job must be terminated using the FTTERM command or the started openFT task must be terminated.

   If the local openFT is then restarted, i.e. the openFT job is loaded with the FJINIT command or openFT is restarted as a "started task" and openFT is then activated with FTSTART, openFT searches for the member in the library hierarchy given above.

If the load module OPENFTCR is not contained in the openFT load library (or a concatenated library), the function "encoded transfer of job data" cannot be used. Depending on your system environment, the following system message is output to the job protocol of openFT after openFT is activated (command FTSTART):

```
CSV003I REQUESTED MODULE OPENFTCR NOT FOUND
```

2.5 Installation of the openFT-AC delivery unit

If openFT is to be used with the FTAC functionality (see section “Administrating and controlling FTAC functions” on page 147), the openFT-AC delivery unit must be installed.

The openFT-AC delivery unit is supplied on CD as a separate order.

The openFT-AC delivery unit is installed with SMP/E as an supplement to the function SYSMOD for openFT described in section “Installing from CD” on page 42. openFT must be available in an SMP/E environment if the openFT-AC delivery unit is to be installed.

openFT (z/OS) including all the additional delivery units is supplied as standard only on CD. Therefore, you must copy the product files from the CD on a Unix or Windows computer and then transfer these to the z/OS computer and unpack them.

Proceed as follows:

1. Perform steps 1 and 2 in section “Transferring files from CD to the z/OS computer and unpacking them” on page 42.
2. Call the procedure FTACLOAD.CLIST:

   ```
   EXEC FTACLOAD
   ```

   This procedure unpacks and loads the XMIT files. This procedure is transferred during step 1 and is located in the TOOLS directory on the CD.

2.5.1 openFT-AC product files

OPENFT.OFA121A.SMPMCS
MCS statements for SMP/E

OPENFT.OFA121A.F1
PO data set containing the following CLIST's:

OFA121A	JCLIN statements for transferring the other product modules from the tape with SMP/E
OPFTAC#1	creates the system and backup files for openFT-AC
OPFTAC#2	extends the SMP/E environment for openFT by the entries needed for openFT-AC
OPFTAC#3	executes the SMP/E statement RECEIVE
OPFTAC#4	executes the SMP/E statement APPLY
OPFTAC#5	executes the SMP/E statement ACCEPT

OPENFT.OFA121A.F2
PO data set containing the following members:

FGMD#AC	contains a Web link to the Release Notice in German
FGME#AC	contains a Web link to the Release Notice in English

OPENFT.OFA121A.F3
Load module library containing the following member:
OPENFTAC Load module for openFT-AC

2.5.2 How to proceed with the installation

The procedure OPFTTEMP.OFA121A.F1 can be used to perform the installation.

The openFT-AC delivery unit can only be installed if the local openFT instance has been fully shut down. You should therefore proceed as follows to perform installation:

1. Shut down openFT using the FTSTOP command.
2. Use the FTTERM command to terminate the started openFT job or terminate the started openFT task.

The following steps are necessary in order to install openFT-AC:

1. Specify the required installation directory or installation prefix by setting the T_BASE variable in the procedure OPFTTEMP.OFA121A.F1.

   The default settings are the user ID and directory in which the temporary installation files are located.

2. Call the following procedures to install openFT-AC:

```
EXEC 'USERID.OPFTTEMP.OFA121A.F1(OPFTAC#1)'
EXEC 'USERID.OPFTTEMP.OFA121A.F1(OPFTAC#2)'
EXEC 'USERID.OPFTTEMP.OFA121A.F1(OPFTAC#3)'
EXEC 'USERID.OPFTTEMP.OFA121A.F1(OPFTAC#4)'
EXEC 'USERID.OPFTTEMP.OFA121A.F1(OPFTAC#5)'
```

   The FTAC command entries created at the time of openFT installation can also be used following openFT-AC installation.

In order to use the FTAC functionality, proceed as follows after the installation:

1. Restart the local openFT instance, i.e. load the openFT job using the FJINIT command or restart openFT as a "started task".
2. Then activate openFT with FTSTART.

2.6 Installing the openFT-FTP delivery unit

If you wish to use openFT with the FTP functionality, you must install the delivery unit openFT-FTP.

The delivery unit openFT-FTP delivery unit is supplied on CD as a separate order.

The openFT-FTP delivery unit is installed using SMP/E as a supplement the "function SYSMOD" for openFT described in section “Installing from CD” on page 42. This means that openFT must be available in an SMP/E environment to be able to install openFT-FTP.

openFT (z/OS) including all the additional delivery units is supplied as standard only on CD. Therefore, you must copy the product files from the CD on a Unix or Windows computer and then transfer these to the z/OS computer and unpack them.

Proceed as follows:

1. Perform steps 1 and 2 in section “Transferring files from CD to the z/OS computer and unpacking them” on page 42.
2. Call the procedure FTFPLOAD.CLIST:

   ```
   EXEC FTFPLOAD
   ```

   This procedure unpacks and loads the XMIT files. This procedure is transferred during step 1 and is located in the TOOLS directory on the CD.

2.6.1 openFT-FTP product files

After unpacking, the following product files are available for openFT-FTP:

OPENFT.OFP121A.SMPMCS
MCS statements for SMP/E

OPENFT.OFP121A.F1
PO data set containing the following CLISTs:

OFP121A	JCLIN statements for transferring the other product modules from the tape with SMP/E
OPFTP#1	creates the system and backup files for openFT-FTP
OPFTP#2	extends the SMP/E environment for openFT-FTP to include the entries required for openFT-FTP
OPFTP#3	executes the SMP/E statement RECEIVE
OPFTP#4	executes the SMP/E statement APPLY
OPFTP#5	executes the SMP/E statement ACCEPT

OPENFT.OFP121A.F2
PO data set containing the following members:
FGMD#FTP contains a Web link to the Release Notice in German
FGME#FTP contains a Web link to the Release Notice in English

OPENFT.OFP121A.F3
load module library containing the following members:
OPENFTP load module for openFT-FTP

2.6.2 How to proceed with the installation

The procedure OPFTTEMP.OFP121A.F1 can be used to perform the installation.

The openFT-FTP delivery unit can only be installed if the local openFT instance has been fully shut down. You should therefore proceed as follows to perform installation:

1. Shut down openFT using the FTSTOP command.
2. Use the FTTERM command to terminate the started openFT job or terminate the started openFT task.

The following steps are necessary in order to install openFT-FTP:

1. Specify the required installation directory or installation prefix by setting the T_BASE variable in the procedure OPFTTEMP.OFP121A.F1.

   The default settings are the user ID and directory in which the temporary installation files are located.

2. Call the following procedures to install openFT-FTP:

```
EXEC 'USERID.OPFTTEMP.OFP121A.F1(OPFTP#1)'
EXEC 'USERID.OPFTTEMP.OFP121A.F1(OPFTP#2)'
EXEC 'USERID.OPFTTEMP.OFP121A.F1(OPFTP#3)'
EXEC 'USERID.OPFTTEMP.OFP121A.F1(OPFTP#4)'
EXEC 'USERID.OPFTTEMP.OFP121A.F1(OPFTP#5)'
```

   The OPENFTP load module contained in the library OPENFT.OFP121A.F3 is installed under the same name as a member of the same library that contains the load modules for openFT.

In order to use the FTP functionality, proceed as follows after the installation:

1. Restart the local openFT instance, i.e. load the openFT job using the FJINIT command or restart openFT as a "started task".
2. Then activate openFT with FTSTART.

If you want to use the openSSL functionality in addition to the FTP functionality offered by the installed unit openFT-FTP, the following line in the batch job or the "started task"

```
//OPENFT EXEC PGM=OPENFT,TIME=1440,
```

must be changed to

```
//OPENFT EXEC PGM=OPENFTSL,TIME=1440,
```

See also “Example of the FJBATCH member” on page 93.

2.7 Startup

A number of steps are required for initial startup of openFT:

– Set the necessary installation parameters with FJGEN, see below.
– Set up the FT parameter library, see page 57.
– Make the OPFT subsystem available, see page 92.
– Specify whether openFT is to run as a job or a started task, see page 92.
– Load and start the openFT load module (see page 96) if openFT is not to be started as a started task.

You can then start, stop and terminate openFT, see page 96.

2.7.1 Setting openFT installation parameters with FJGEN

You use the TSO procedure FJGEN from the library OPENFT.CLIST to set up a new openFT instance or, subsequently, to modify the parameter settings of existing instances. In your openFT system, instances are identified via their instance names which you specify in INSTANCE NAME during FJGEN processing. FJGEN expects the load modules of openFT to be specified in the libraries OPENFT.LOAD and OPENFT.NCLOAD.

FJGEN runs a dialog to query the parameters to be defined during the first installation step. In particular, the qualifier under which the openFT administrative files and the instance-specific FT procedure library are stored is defined here. (OPENFT QUALIFIER).

The FJGEN command is required even if the installation parameters have been set using a file (see section “Setting up the FT parameter library” on page 57). If openFT is to run as a started task, the FT administrator must create the start procedure himself and specify the corresponding values for the installation parameters (see section “openFT as a job or started task” on page 92). Here also, however, the FJGEN command is required to create further command procedures and create instance-specific files.

The FJGEN command can also be used to change the installation parameters of an openFT instance. The named procedures are then regenerated in the instance-specific procedure library. The changes become effective the next time the installation-specific batch job is started with FJINIT from the <openFT qualifier>.<inst>.CLIST library.

If openFT is running as a started task, the FT administrator is responsible for making the corresponding changes in the start procedure, see section “openFT as a job or started task” on page 92.

When an instance-specific procedure library is concatenated, you can obtain information about the current values of the installation parameters using the FJGENPAR command from the <openFT qualifier>.<inst>.CLIST library.

A detailed description of the FJGEN command and examples are given on page 104.

2.7.2 Setting up the FT parameter library

You can use the FT parameter library to tailor openFT to the specific requirements of your installation. You can store the following information in the members of this library:

- openFT Installation parameters: Some installation parameters are also specified in the FJGEN command, see page 104 and page 56; installation parameters entered in the FT parameter library take precedence over those specified in FJGEN.)
- Definitions of the users (user IDs) who possess FT or FTAC administrator rights
- Job cards for printing the result list and preprocessing, postprocessing and follow-up processing.
- Address information from an older openFT version that is still to be used in openFT V12.
- Details on the use of file-specific character sets (see section “Structure of the member FNAMECTB” on page 87 and section “Administering code tables” on page 125)
- Installation parameters required when openFT-AC is used
- Specifications for creating diagnostics information.

Format and name of the FT parameter library

The information from the FT parameter library is stored as text in members of a PO or PDSE dataset. By default, the library must be created under the following name:

<openft qualifier>.<inst>. PARM

The OPENFT QUALIFIER (<openft qualifier>) and the INSTANCE NAME(<inst>) are defined using FJGEN (see page 104).

The following also applies:

- It is advisable to create this data set with the record format F or FB and a record length of 80. openFT fills records with a length of less than 80 in order to make them 80 characters long. Records whose length exceeds 80 characters are truncated after the 80th character. This can lead to errors if invalid job cards are created as a result, particularly for the members TSOJOB, TSOVVJOB, TSONVJOB, TSOVFJOB, JCLJOB and PRTJOB (see below).
- The members of the FT parameter library may **not** contain line numbering. Please observe this rule when creating or editing the text contained in the members. (If you use the PDF editor, e.g. via the menu interface for the FT administrator, you must therefore set NUMBER OFF in your EDIT profile)

Elements of the FT parameter library

PARM:
Installation parameters for openFT. The structure of this member is described on page 60.

FTADM:
List of users with FT administration authorization. The structure of this member is described on page 70.

FTACADM:
List of users with FTAC administration authorization. The structure of this member is described on page 70.

PRTJOB:
Job cards for printing the result list. The structure of this member is described on page 72.

JCLJOB:
Job cards for the follow-up job which is created by openFT if the follow-up processing consists of one or more JCL statements. The structure of this member is described on page 72.

TSOJOB:
Job cards for the follow-up job which is created by openFT if the follow-up processing consists of one or more TSO commands. The structure of this member is described on page 72.

TSOVVJOB:
Job cards for the preprocessing job generated by openFT if one or more TSO commands have been requested as preprocessing commands in an FT request. The structure of this member is described on page 72.

TSOVFJOB:
Job cards for the preprocessing job generated by openFT if the "ftexec" command is issued for the z/OS system. The structure of this member is described on page 72.

TSONVJOB:
Job cards for the postprocessing job generated by openFT if one or more TSO commands have been requested as postprocessing commands in an FT request. The structure of this member is described on page 72.

SUCCMSG:
Text of the asynchronous message which openFT issues as a result of successful file transfer to one or several consoles. This member is evaluated only if a valid specification has been made for the SUCC_MSG keyword in the FTMSPPAR member. The structure of this member is described on page 84.

FAILMSG:
: Text of the asynchronous message which openFT issues as a result of unsuccessful file transfer to one or several consoles. This member is evaluated only if a valid specification has been made for the FAIL_MSG keyword in the FTMSPPAR member. The structure of this member is described on page 84.

FNAMECTB:
: Information on of which file-specific code character sets (see section “Administering code tables” on page 125) openFT is to use and which files are to be coded with the relevant character set. The structure of the member is described on page 87.

FTACPAR:
: Installation parameter needed when openFT-AC is used. The structure of the member is described on page 91.

DIAGPAR:
: Specifications for creating diagnostics information. The structure of the member is described on page 174.

The members PARM, PRTJOB, JCLJOB, TSOJOB, TSOVVJOB, TSONVJOB, TSOVFJOB, SUCCMSG, FAILMSG, FTADM, FTACADM, FNAMECTB, FTACPAR and DIAGPAR are read and evaluated when the openFT load module is started, i.e. when the openFT batch job is started (with FJINIT) or when the started task commences. Errors in accessing a member are not reported. openFT acts as though the member concerned were not present. In this case the appropriate default values are used, if any are available.

Any updates to the FTADM, FTACADM, DIAGPAR and FNAMECTB members can be read in during system operation using the FTUPDPAR command (see page 117).

2.7.2.1 Structure of the PARM member

You can specify installation parameters for openFT in this element.

The description of the FJGEN command (page 104) shows the structure of the batch job which is required for openFT to run as a background task (FJBATCH member of the FT procedure library). This job remains unchanged even if you specify installation parameters in the PARM member of the FT parameter library. In particular, the string containing the start parameters in the batch job remains unchanged. If, however, you also specify a corresponding parameter in the PARM member, openFT uses this value; the value from the start parameters is not used in this case.

Each line of the PARM member can contain exactly one parameter in the form "keyword=value". No blanks may be inserted between "keyword", "=" and "value". Below is a list of the keywords which may be used.

Keywords:

DESTVOL=
: Definition of the volume for local receive files which do not exist and for files used to store the result lists (LISTING=*LISTFILE). Exactly 6 characters. See also the description of the DESTUNIT parameter.

DESTUNIT=
: Definition of the unit/groupname for local receive files which do not exist and for files used to store result lists (LISTING=*LISTFILE). Maximum 6 characters.

  If you specify a value for only one of the parameters DESTVOL and DESTUNIT, openFT assigns blanks to the other. If you do not specify a value for either of the parameters DESTVOL and DESTUNIT, receive files which do not exist and files for result lists are created on the default volume (system-specific).

UNLOADVOL=
: Definition of the volume for temporary PS data sets with the suffix ".U" to which the entire PO/PDSE data sets are transferred (in "unloaded" format) prior to file transfer. Exactly 6 characters. See also the description of the UNLOADUNIT parameter.

UNLOADUNIT=
: Definition of the unit/groupname for temporary PS data sets with the suffix ".U" to which entire PO/PDSE data sets are transferred (in "unloaded" format) prior to file transfer. Maximum 6 characters.

  If you specify a value for only one of the parameters UNLOADVOL and UNLOADUNIT, openFT assigns blanks to the other. If you do not specify a value for either of the parameters UNLOADVOL and UNLOADUNIT, the temporary PS data sets are created on the default volume (system-specific).

  In order to transfer entire PO/PDSE data sets, openFT must be APF-authorized (see section “openFT privileges” on page 38).

PODIR=

Number of directory blocks which are to be reserved by openFT when creating a PO data set. Maximum value: 32767 (default: 20).

DSTYPEDEF=

Default value for file organization if the receive file of a file transfer request is to be created as a sequential file whose file organization is not precisely defined by the structure of the send file (if homogeneous systems are used as of V10 partners) or the file name (see the manual "openFT (z/OS) - Command Interface").
Possible Values:

PS A "physical sequential data set" (PS data set) is created.

VSAM A VSAM file of type "entry sequenced" is created.

If no value or an invalid value is specified for the DSTYPEDEF parameter, the default value PS applies.

LIBTYPEDEF=

Default value for the file organization if a data set (library) divided into members is to be created for a file transfer request in the receive system and the file organization is not precisely defined by the structure of the send file (if a complete PO or PDSE data set is transferred as of V10 partners) or the file name (see the manual "openFT (z/OS) - Command Interface").
This value is important

- when a member is the receive file, the associated data set does not yet exist and the type (PO/PDSE) has not been specified (prefix :L: or no prefix),
- when a complete PO or PDSE data set is the receive file, the type (PO/PDSE) has not be specified precisely (prefix :L:).

Possible values:

NONE openFT does not specify the file organization, the system defaults therefore apply:
  - if the IBM software product DFSMS (Data Facility System-Managed Storage) is not installed, PO is the default
  - if DFSMS is active: definition of the default by the ACS routine or SYS1.PARMLIB (please ask your z/OS system administrator)

PO A "partitioned organized data set" (PO data set) is created.

PDSE An attempt is made to create a "partitioned organized data set extended" (PDSE data set). This is only possible if the IBM software product DFSMS is installed and the parameters PDSESTORC, PDSEMGMTC and PDSEDATAC (see below) have been set correctly.

If no value or an incorrect value is specified for LIBTYPEDEF, the default value NONE applies.

PDSESTORC=

SMS storage class for PDSE data sets (refer to the literature on the IBM software product DFSMS for further details).
Maximum 8 characters; valid name of an SMS storage class.
The value is only used if there are no settings concerning the SMS storage class on your system (please ask your z/OS system administrator).

Please observe the description of the PDSEDATAC parameter.

PDSEMGMTC=

SMS management class for PDSE data sets (refer to the literature on the IBM software product DFSMS for further details).
Maximum 8 characters; valid name of an SMS management class.
This value is only used if there are no settings concerning the SMS management class on your system (please ask your z/OS system administrator).

Please observe the description of the PDSEDATAC parameter.

PDSEDATAC=

SMS data class for PDSE data sets (refer to the literature on the IBM software product DFSMS for further details).
Maximum 8 characters; valid name of an SMS data class.
This value is only used if there are no settings concerning the SMS data class on your system (please ask your z/OS system administrator).

The parameters PDSESTORC, PDSEMGMTC and PDSEDATAC only become effective if a PDSE data set is to be newly generated as a receive file (either because a complete PDSE file set was specified as the receive file or because a PDSE member is a receive file, but the relevant PDSE data set does not exist yet).

The parameters all only become effective if your system does not contain any specifications for the relevant SMS class.

As a rule, you should not specify any of these parameters; the settings made in your system will then apply. Exception: if no settings are made in your system for any of the SMS classes, you must specify at least one of the parameters PDSESTORC, PDSEMGMTC and PDSEDATAC, otherwise openFT will not be able to generate a PDSE data set.

An invalid specification for one of the parameters, which will become effective (because there are no default settings for the relevant SMS class in your system) will cause those transfer requests to fail for which a new PDSE data set has to be generated on the receive system. Further details are contained in the manual "openFT (z/OS) - Command Interface").

POSTORC=
SMS storage class for PO data sets (refer to the literature on the IBM software product DFSMS for further details).
Maximum 8 characters; valid name of an SMS storage class.
This value is only used if there are no settings concerning the SMS storage class on your system (please ask your z/OS system administrator).

Please observe the description of the PODATAC parameter.

POMGMTC=
SMS management class for PO data sets (refer to the literature on the IBM software product DFSMS for further details).
Maximum 8 characters; valid name of an SMS management class.
This value is only used if there are no settings concerning the SMS management class on your system (please ask your z/OS system administrator).

Please observe the description of the PODATAC parameter.

PODATAC=
SMS data class for PO data sets (refer to the literature on the IBM software product DFSMS for further details).
Maximum 8 characters; valid name of an SMS data class.
This value is only used if there are no settings concerning the SMS data class on your system (please ask your z/OS system administrator).

If PO data sets are to be created as SMS-managed data sets (prerequisite for this is that the IBM software product DFSMS is installed), the same in essence applies for parameters POSTORC, POMGMTC and PODATAC as for parameters PDSESTORC, PDSEMGMTC and PDSEDATAC (see description of parameter PDSEDATAC on page 62).

JOB_MSGCLASS=
Message class of the follow-up processing job, the preprocessing job and the postprocessing job. This is the default value for the JOB statement parameter MSGCLASS= if the members TSOJOB, JCLJOB, TSOVVJOB, TSOVFJOB or TSONVJOB contain no relevant entry. Exactly 1 character (default value: A).

LST_MSGCLASS=
Message class of the job for printing the result list. This is the default value for the JOB statement parameter MSGCLASS= if the member PRTJOB contains no relevant entry. Exactly 1 character (default value: A).

LISTPARM=
Requests a result list in the local system. This listing is generated for the user for whom file transfer is performed. The value specified here applies to the NCOPY command if LISTING=*STD is specified there.

Possible values: LISTFILE, SYSLST and NONE, meaning see NCOPY command.

JOB_JOBCLASS=

Job class of the follow-up processing job, the preprocessing job, the postprocessing job, and the print job. This is the default value for the JOB statement parameter CLASS= if the members TSOJOB, JCLJOB, TSOVVJOB, TSOVFJOB, TSONVJOB or PRTJOB contain no relevant entry.
Exactly 1 character (default value: A).

NABVOLUME=

The volume on which the request file, the partner list, the operating parameter file, the logging file and the FTAC file are to be located (see section “Internal openFT data sets” on page 193).
Exactly 6 characters.

If the corresponding files are SMS managed, the specifications for Volume and Unit may have no effect under certain circumstances. If the files are not SMS managed, an "SMS managed volume" must not be specified here.

See also the description of the NABUNIT parameter.

NABUNIT=

Definition of the unit/groupname of the volume on which request file, the partner list, the operating parameter file, the logging file and the FTAC file is to be set up. Maximum 6 characters.

If you specify a value for only one of the parameters NABVOLUME and NABUNIT, openFT assigns blanks to the other.

If you do not specify a value for either of the parameters NABVOLUME and NABUNIT, the values of DMP_VOLUME and DMP_UNIT (see below) are assumed. Either or both of these values may in turn have been taken from the values specified for VOLUME/UNIT in the FJGEN command.

LOGFILE_2ND_Q=

The second level qualifier for creating the names of the components of the logging file (see section “Internal openFT data sets” on page 193).
Up to 18 characters (default: <inst>.SYSLOG, where <inst> is the name of the openFT instance). For the sake of clarity, the name should always start with the instance name followed by a period.

Depending on the length of the "second level qualifier", the timestamp in the log file name can be truncated or omitted entirely. In such cases, the possibility of changing log files using the FTMODOPT command is either restricted or unavailable.

This name must be specified in partially qualified form, i.e. with no "first level qualifier" or single quotes. openFT prefixes this name with the OPENFT QUALIFIER specified in FJGEN.

LOGFILE_SIZE_RC=

Initial size of the logging file (number of logging records).
Maximum value: 16777215 (default: 10000).

openFT uses this value as the primary allocation when creating the VSAM cluster which is part of the logging file. For the secondary allocation, the value is halved.

The specified maximum value is the program-technical limit. When choosing a value for the initial size of the logging file, the actually available storage space needs to be taken into consideration. Note that the logging file is created on the same data volume as the request file, the partner list and (if FTAC is installed) the FTAC file (see section “Internal openFT data sets” on page 193).

DMP_VOLUME=

Definition of the volume on which openFT creates the dump and trace files. Exactly 6 characters (default: value specified for VOLUME in the FJGEN command; see the description of the FJGEN command, page 104). You can also use "DMP_VOLUME= " to specify that the value specified for VOLUME in the FJGEN command is not used when the dump and trace files are created. See also the description of the DMP_UNIT parameter.

DMP_UNIT=

Definition of the unit of the volume on which openFT creates the dump and trace files. Maximum 6 characters (default: value specified for UNIT in the FJGEN command (page 104).

You can also use "DMP_UNIT= " to specify that the value specified for UNIT in the FJGEN command is not used when the dump and trace files are created.

If, after evaluation of the specifications for FJGEN (VOLUME/UNIT) and the specifications made here, there is no value for DMP_VOLUME or for DMP_UNIT, openFT uses the UNIT name DASD. This UNIT name must then be defined in the system.

Examples

VOLUME/UNIT (FJGEN)	**FTMSPPAR**	**Result**
VSN123/SYSDA	DMP_VOLUME=VSN456	DMP_VOLUME = VSN456 DMP_UNIT = SYSDA
VSN123	DMP_UNIT=SYSDA	DMP_VOLUME = VSN123 DMP_UNIT = SYSDA
/	DMP_VOLUME=VSN456 DMP_UNIT=SYSDA	DMP_VOLUME = VSN456 DMP_UNIT = SYSDA
VSN123/SYSDA	DMP_VOLUME=	DMP_VOLUME = no value DMP_UNIT = SYSDA
VSN123	DMP_VOLUME=	DMP_VOLUME = no value DMP_UNIT = DASD (!)

VOLUME/UNIT (FJGEN)	FTMSPPAR	Result
/	no specifications	DMP_VOLUME = no value DMP_UNIT = DASD (!)

ROUTCDE=
Routing code of one console to which the openFT asynchronous messages are to be output. (Note openFT uses the WTO macro to output these messages to the console.) Valid values: 1 through 128. Invalid values are ignored and no message is issued.

SMF_RECORD_TYPE=
Type of the accounting record written by openFT to the SMF file. Valid values:

128 through 255
: For each transfer request accepted, an accounting record of the specified type is written to the SMF file, provided that SMF is active. The structure of the accounting records is described in the appendix.

0
: No accounting records are written to the SMF file. (Default; values outside the valid range are interpreted as 0.)

SMF_ADM_AREA=
Installation-specific text written by openFT to the FT administrator area of the accounting records (see the description of the accounting record structure in the section “Accounting records” on page 185). This text may be up to 40 characters long. Default: blanks.

In order to enter SMF accounting information, openFT must be APF-authorized (see section “openFT privileges” on page 38).

SUCC_MSG=
Specifies when an asynchronous message indicating successful file transfer is to be issued. Valid values:

IN
: An asynchronous message indicating successful file transfer is output only for transfer requests which were submitted in a remote system.

OUT
: An asynchronous message indicating successful file transfer is output only for transfer requests which were submitted in the local system.

BOTH
: An asynchronous message is output for all transfer requests following successful file transfer.

Invalid values are ignored and no error message is issued. In this case, no asynchronous message is output following successful file transfer (default value).

The destination for output of the asynchronous message after successful file transfer is controlled by the keyword ENDMSG_ROUTCDE (see below).

The message text for the asynchronous message following successful file transfer can be defined in the SUCCMSG member of the FT parameter library; otherwise, openFT uses a standard text (see page 84).

In order to output asynchronous messages following termination of a request, openFT must be APF-authorized (see section “openFT privileges” on page 38).

FAIL_MSG=

Specifies when an asynchronous message indicating unsuccessful file transfer is to be output. Valid values:

IN An asynchronous message indicating unsuccessful file transfer is output only for transfer requests which were submitted in a remote system.

OUT An asynchronous message indicating unsuccessful file transfer is output only for transfer requests which were submitted in the local system.

BOTH An asynchronous message is output for all transfer requests following unsuccessful file transfer.

Invalid values are ignored and no error message is issued. In this case, no asynchronous message is output following unsuccessful file transfer (default value).

The destination for output of the asynchronous message after successful file transfer is controlled by the keyword ENDMSG_ROUTCDE (see below).

The message text for the asynchronous message following unsuccessful file transfer can be defined in the FAILMSG member of the FT parameter library; otherwise, openFT uses a standard text (see page 84).

In order to output asynchronous messages following termination of a request, openFT must be APF-authorized (see section “openFT privileges” on page 38).

ENDMSG_TO_TSO=

Switch for controlling the output of asynchronous messages to a TSO terminal at the end of a job. The output is made to the terminal of the TSO user whose user ID was specified in the TRANSFER-ADMISSION. The messages are only output for jobs issued locally; You can find them in the appendix as of page 206.
Possible values:

YES (Default): The asynchronous messages are output.

NO (Or invalid value): The asynchronous messages are not output.

The asynchronous messages output to a TSO terminal at the end of a job also appear in the openFT job protocol (see page 191).

ENDMSG_ROUTCDE=

Routing code of one console to which the asynchronous messages are to be output at the end of a job. The cases in which an asynchronous message is output at the end of a job is controlled by the keywords SUCC_MSG and FAIL_MSG (see above).
Possible values: 1 to 128. Invalid values are ignored and no message output.

If the keyword is missing or assigned no (or an invalid) value, no output is made to the console.

The asynchronous messages output to a console at the end of a job are assigned a key (FJM2100 for the message following successful file transfer); this makes it possible to process these messages with NetView. The messages also appear in the job protocol openFT in this form (see page 191).

TCP_USERID=

Name of the TCP/IP address space. If the name of the TCP/IP address space is not TCPIP (default), you must specify it here. Message FTR4055 can indicate that the name of the TCP/IP address space has not been specified correctly. Ask your z/OS system administrator.

Up to 8 characters (default: TCPIP).

MSG_CRYPT=

Optionally, it is possible to encrypt the messages from the openFT dialog tasks for the purposes of internal communications with the openFT subsystem. Commands are always encrypted. The mechanism employed is the same as for the encryption of the request description data.
Valid values:

Y	Messages are encrypted.
N	Messages are not encrypted (default).

CMD_TRANS=

You use this switch to define the transport protocol to be used to connect the dialog tasks to the openFT subsystem. If openFT implicitly recreates the parameter library then CMD_TRANS=TCP is preset.
Valid values:

VTAM	Communication is performed via VTAM.
TCP	Communication is performed via TCP (default)

OPENFT_SVC=

The openFT subsystem administers all the running instances and encrypts or decrypts all the commands, messages (optional, see the MSG-CRYPT parameter) and connection data. The portal to the subsystem is implemented via SVC 109 with "extended code 211". This ESR SVC code is defined using OPENFT-SVC. If "extended Code 211" is already used for a different purpose in your system, you can use the LINK procedure
LINKIGX from the SAMPLES library in order to utilize your extended code. For reasons of security, OPENFT-SVC should be set to a valid value.

USER_INACT_TIME

This specification defines a maximum idle time (in minutes) before a connection between the user TSO interface and the openFT subsystem will be terminated for security reasons.
Valid values:

0 ..30 Time specification for the maximum idle time in minutes.
(Default: 5 minutes)

PSSTORC=

SMS storage class for PS datasets. For further details, see the documentation on the IBM software product DFSMS.

Up to 8 characters; valid name of an SMS storage class.
This value only has any effect of there is no default specification for the SMS storage class in your system (ask your z/OS system administrator).
Refer also to the description of the PSDATAC parameter.

PSDATAC=

SMS data class for PS datasets. For further details, see the documentation on the IBM software product DFSMS.

Up to 8 characters; valid name of an SMS data class.
This value only has any effect of there is no default specification for the SMS data class in your system (ask your z/OS system administrator).

The PSSTORC and PSDATAC parameters only take effect if a new PS dataset is to be created as the receive file. These parameters are also valid for creating trace files.

DEFFSIZE=

Size of a secondary allocation for the receive file if the size of the send file is unknown. DEFFSIZE is specified in bytes. In this case, the primary allocation is approximately one tenth of this value. If this specification is omitted, DEFFSIZE=2621440 is taken. The DEFFSIZE parameter also influences the size of the temporary file for data output that is used during preprocessing and/or preprocessing with FTEXEC. If, for example, you want to retrieve large data volumes from z/OS at an external platform using the FTEXEC command or GUI available there then you should set the DEFFSIZE

parameter to a sufficiently large value (see also the PALC and SALC parameters during preprocessing with FTEXEC on page 74). For further details, refer to the section "File types - z/OS files" in the User Guide.

MAXALLOC=
Maximum size of file allocations (both primary and secondary). MAXALLOC is specified in megabytes. The default value is 1024, and the (theoretical) maximum value is 32767.

Example of the member PARM

```
DESTVOL=TSOOOO
DESTUNIT=SYSDA
UNLOADVOL=TSOOOO
UNLOADUNIT=SYSDA
DSTYPEDEF=PS
LIBTYPEDEF=PO
JOB_MSGCLASS=X
LST_MSGCLASS=X
SUCC_MSG=BOTH
FAIL_MSG=BOTH
ENDMSG_TO_TSO=YES
TCP_MYPORT=1100
LOGFILE_2ND_Q=OPENFTLG
```

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). See page 69 for details on DEFFSIZE.

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 page 70. If the allocations calculated using the method described above do not exceed this threshold, MAXALLOC is of no significance.

2.7.2.2 Structure of the members FTADM and FTACADM

The members FTADM and FTACADM contain all the users (user IDs) who possess FT or FTAC administrator authorization. Each entry must start on a new line in column 1. User ID groups that differ only in the associated suffix and which all possess the same authorizations can be combined using wildcards "*". For example, the user IDs XORG001, XORG002 and XORG003 can be represented by a single entry XORG*.

If you enter administration commands at the console or use Netview then a pseudo-entry *Console* must be set up in these members.

In FTACADM, it is possible to assign FTAC administrators what FTAC considers to be "system administrator rights". The restrictions applying to the setup and import of admission profiles for external user IDs do not apply to these administrators. To assign this privilege, enter SU after the user ID in column 10/11.

WARNING!
FTAC administrators with the "SU privilege" can set up appropriate admission profiles allowing them to access the files belonging to any user ID and, in this way, circumvent any protection policies that may be in place! For this reason, it is necessary to treat write access rights to the FT parameter library with considerable care.

If the PARM library does not exist at the time the FJGEN command is called, openFT creates the members FTADM and FTACADM during FJGEN execution. These are then assigned the OPENFT USER ID and the pseudo-entry "Console". The members may be modified (e.g. other user IDs may be entered in them). Changes take effect the next time openFT is loaded or when the FTUPDPAR command is issued. FTADM and FTACADM may each contain up to 100 entries.

2.7.2.3 Structure of the members PRTJOB, JCLJOB, TSOJOB, TSOVVJOB, TSOVFJOB and TSONVJOB

Each of these members consist of prototype statements which openFT uses if it is creating an appropriate job internally. openFT does not check the syntax of these prototype statements. Sample members are supplied in the library SAMPLES that is delivered with openFT.

Each record contains exactly one job card (or continuation card). A maximum of 32767 records are evaluated for each member.

The following variables can be used in these prototype statements. openFT replaces these symbolic parameters with the current values:

JOBP
: Job name prefix, identical to USID if the user ID does not exceed 7 characters in length. Otherwise, the last character is removed.

USID
: User ID from TRANSFER-ADMISSION (for TSOVVJOB, TSOVFJOB and TSONVJOB) or from PROCESSING-ADMISSION (for JCLJOB and TSOJOB).
Maximum 8 characters, in accordance with IBM conventions.

ACCN
: "accounting information" from TRANSFER-ADMISSION (PRTJOB, TSOVVJOB, TSOVFJOB and TSONVJOB) or from PROCESSING-ADMISSION (for JCLJOB and TSOJOB).
Maximum 40 characters, in accordance with IBM conventions.

PASS
: Password from TRANSFER-ADMISSION (PRTJOB, TSOVVJOB, TSOVFJOB and TSONVJOB) or from PROCESSING-ADMISSION (for JCLJOB and TSOJOB).
Maximum 8 characters, in accordance with IBM conventions.

OWID
: Owner of the FT request, i.e. the user ID under which the transfer job was created.
Maximum 8 characters according to IBM conventions.

  This variable is replaced only in the system where the transfer request was issued. It is eliminated in the remote system.

PGRN
: „programmer's name" as specified a subcommand in the relevant command string for follow-up processing with the keyword PGRN= (see the manual "openFT (z/OS) - Command Interface"). The PRTJOB member also assigned the value from the relevant command string for follow-up processing, i.e. from the command string in the

SUCCESS-PROCESSING on successful processing and from the command string in the FAILURE-PROCESSING if processing failed.
Maximum 20 characters in accordance with IBM conventions.

If there is no value for "programmer's name" and, after replacement of the PGRN variables inside a JOB Statement, it is established that the corresponding card has no other data except for the JCL identifier „//" and comma (separator for parameters), this card is ignored, i.e. there is no execution. This is in keeping with the recommendation in the IBM literature (JCL Reference) not to mark this missing parameter with a comma.

TRID
FT transfer ID.
Maximum 10 characters (value range 1..2147483639) in accordance with openFT conventions.

RLFN
Name of the file in which the result list is stored (PRTJOB). Maximum of 56 characters, in accordance with IBM conventions (the maximum length is obtained from the specified structure of this file name; refer to the manual "openFT (z/OS) - Command Interface" for further details).
This variable can be used in the member PRTJOB; in other members it is removed.

RLFP
Temporary file to which the preprocessing operation outputs your data (TSOVVJOB and TSOVFJOB). Maximum 38 characters in accordance with IBM conventions. This variable can only be used in the members TSOVVJOB and TSOVFJOB. In other members, it is removed.

RLFF
Temporary file to which the preprocessing operation of an ftexec command outputs its error messages. Maximum 38 characters in accordance with IBM conventions. This variable can only be used in the member
TSOVFJOB. In other members, it is removed.

RLFT
Temporary file to which the preprocessing operation of an ftexec command issued in a Unix or Window partner system outputs its TSO messages. Maximum 38 characters in accordance with IBM conventions. This variable can only be used in the member TSOVFJOB. In other members, it is removed.

CONN
Name of the file containing the key for the connection to the FT subsystem. Maximum 36 characters in accordance with IBM conventions. By default, this file is created under <openft qualifier>.<inst>.CONN.This variable can only be used in the members TSOJOB, TSONVJOB, TSOVVJOB and TSOVFJOB. In other members, it is removed.

NCLO
: Name of the file that contains the openFT commands. Maximum 36 characters in accordance with IBM conventions. By default, this file is created under OPENFT.NCLOAD. This variable can be used in the members TSOJOB, TSONVJOB, TSOVVJOB and TSOVFJOB. It is eliminated from other members.

PALC
: Size in kilobytes of the primary allocation for the output file that is used temporarily during preprocessing with the FTEXEC command.
Default value: 256
If you want to change this value then you must modify the value for DEFFSIZE in the PARM file (see page 69). Modifying the value of DEFFSIZE not only affects the size of the temporary file during preprocessing with FTEXEC but also more generally affects the temporary files used during preprocessing for interim data output.
Example: If you double the default value for DEFFSIZE (2621440) then the value 256 is doubled.

SALC
: Size in kilobytes of the secondary allocation for the output file that is used temporarily during preprocessing with the FTEXEC command.
Default value: 2560
If you want to change this value then you must modify the value for DEFFSIZE in the PARM file (see page 69). Modifying the value of DEFFSIZE not only affects the size of the temporary file during preprocessing with FTEXEC but also more generally affects the temporary files used during preprocessing for interim data output.
Example: If you double the default value for DEFFSIZE (2621440) then the value 2560 is doubled.

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 fill characters.

Note

If a follow-up processing job of the type ALLOC DSNAME (...) was specified, openFT also replaces the variables in this job before passing it to the Internal Reader.

When creating sample instructions, the requirements of the relevant z/OS installation for executable jobs must be observed. As a rule, the JOB statement requires a valid user ID, valid accounting information and a valid user password. These values can be taken from the following sources:

– They are specified by the user in the NCOPY command (for PRTJOB, TSOVVJOB, TSOVFJOB and TSONVJOB in the TRANSFER-ADMISSION, for JCLJOB and TSOJOB in the PROCESSING-ADMISSION).

– If FTAC is used, the specifications for PROCESSING-ADMISSION can also be defined within an admission profile. The admission to perform preprocessing and postprocessing is defined via the specifications for the TRANSFER-ADMISSION.

– They are contained in the JOB statements in the PRTJOB, JCLJOB,TSOJOB, TSOVVJOB, TSOVFJOB and TSONVJOB members, i.e. no variables are used for the user ID, accounting information and user password. In this case, the specifications apply for all jobs.

These notes also apply to default jobs created by openFT if the TSOJOB, TSOVVJOB, TSOVFJOB and TSONVJOB members do not exist.

The examples below reflect the default structure of the jobs created by openFT. Deviations specific to the computer center can be implemented in the members PRTJOB, JCLJOB, TSOJOB, TSOVVJOB, TSOVFJOB and TSONVJOB.

Example of the member PRTJOB

```
//&JOBP##P JOB &ACCN########################,
//             &PGRN################,
//             MSGCLASS=X,
//             CLASS=C,
//             USER=&USID###,PASSWORD=&PASS###
//PRTJOB    EXEC PGM=IEBPTPCH
//SYSPRINT  DD DUMMY
//SYSUT1    DD DSN=&RLFN########################################,
//          DISP=(SHR,DELETE),
//          DCB=(RECFM=FB,LRECL=134,BLKSIZE=2546)
//SYSUT2    DD SYSOUT=A,DCB=(LRECL=134)
//SYSIN     DD *
     PRINT PREFORM=A
/*
//
```

In this example the user ID, account number and password are inserted by openFT from the user's TRANSFER-ADMISSION entry. For the PGRN variable, the value specified by the user with the keyword PGRN= as subcommand in the command string for follow-up processing is used (for more detail, please refer to the manual "openFT (z/OS) - Command Interface"). If no value exists for „programmer's name", this card is ignored, i.e. there is no execution, since it contains no other data except for the JCL identifier „//" and comma (separator for parameters). If job processing is successful, the value is taken from the command string in the SUCCESS-PROCESSING parameter; if not, it is taken from the command string in the FAILURE-PROCESSING parameter. If no value exists for „programmer's name", this card is ignored, i.e. there is no execution, since it contains no other data except for the JCL identifier „//" and comma (separator for parameters). In addition, the name of the file with the result list is inserted (the field &RFLN##...## provided for this name should be 32 characters long). openFT then initiates this job.

Example of the member JCLJOB

```
//&JOBP##N JOB &ACCN######################,
//         MSGCLASS=X,
//         CLASS=C,
//         REGION=2M,
//         USER=&USID###,
//         PASSWORD=&PASS###
//JOBLIB    DD  DSN=&USID###.PROCLIB,DISP=SHR
```

In this example the user ID, the account number and the password are inserted by openFT from the user's PROCESSING-ADMISSION entry. For the PGRN variable, the value specified by the user with the keyword PGRN= as subcommand in the command string for follow-up processing is used (for more detail, please refer to the manual "openFT (z/OS) - Command Interface"). If no value exists for „programmer's name", this card is ignored, i.e.

there is no execution, since it contains no other data except for the JCL identifier „//" and comma (separator for parameters).

The JCL statements specified by the user in the NCOPY command as follow-up processing are added by openFT after the prototype statement "//JOBLIB DD ...". openFT then initiates this job.

Example of the member TSOJOB

```
//&JOBP##N  JOB  &ACCN###################################,
//          &PGRN###############,
//          MSGCLASS=X,
//          CLASS=C,
//          NOTIFY=&USID###,
//          USER=&USID###,
//          PASSWORD=&PASS###,
//          REGION=0M
//          EXEC PGM=IKJEFT01
//OPENFT    DD DSN=&CONN########################################,
//             DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO########################################,
//             DISP=(SHR,KEEP)
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
```

In this example the user ID, the account number and the password are inserted by openFT from the user's PROCESSING-ADMISSION entry. For the PGRN variable, the value specified by the user with the keyword PGRN= as subcommand in the command string for follow-up processing is used (for more detail, please refer to the manual "openFT (z/OS) - Command Interface"). If no value exists for „programmer's name", this card is ignored, i.e. there is no execution, since it contains no other data except for the JCL identifier „//" and comma (separator for parameters). The TSO commands specified by the user in the NCOPY command as follow-up processing, as well as the end of data terminator "/*", are added by openFT after the prototype statement "//SYSTSIN DD *". openFT then initiates this job.

Example of the member TSOVVJOB

```
//&JOBP##N  JOB  &ACCN###################################,
//          MSGCLASS=X,
//          CLASS=C,MSGLEVEL=(1,1),
//          USER=&USID###,
//          PASSWORD=&PASS###,
//          REGION=0M
//STEP0     EXEC PGM=IKJEFT01,
//          COND=(0,NE)
//OPENFT    DD DSN=&CONN#########################################,
//             DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO#########################################,
//             DISP=(SHR,KEEP)
//SYSTSPRT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSTSIN   DD *
FTATTP &PRID
```

In this example, the user ID, account number and password are inserted by openFT from the user's TRANSFER-ADMISSION. openFT allocates the appropriate files for the variables CONN, NCLO and RLFP.

If %TEMPFILE was not specified during preprocessing, openFT extends the job as follows:

```
//*****************************************************
//STEP1     EXEC PGM=IEFBR14
//STDOUT    DD DSN=&RLFP#################################,
//          DISP=(NEW,CATLG,DELETE),
//          DCB=(DSORG=PS,BLKSIZE=1536,RECFM=VB),
//          UNIT=SYSDA,SPACE=(1,(256,2560)),AVGREC=K
//*****************************************************
//IFBAD     IF STEP0.RC=0 THEN
//STEP2     EXEC PGM=IKJEFT01,
//          COND=(0,NE)
//OPENFT    DD DSN=&CONN#################################,
//          DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO#################################,
//          DISP=(SHR,KEEP)
//SYSTSPRT  DD SYSOUT=*
//SYSPRINT  DD DSN=*.STEP1.STDOUT,DISP=(MOD,KEEP,DELETE)
//SYSTSIN   DD *
//IFBADEND  ENDIF
```

If %TEMPFILE was specified during preprocessing, openFT extends the job as follows:

```
//IFBAD     IF STEP0.RC=0 THEN
//STEP2     EXEC PGM=IKJEFT01,
//          COND=(0,NE)
//OPENFT    DD DSN=&CONN##################################,
//          DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO##################################,
//          DISP=(SHR,KEEP)
//SYSTSPRT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSTSIN   DD *
//IFBADEND  ENDIF
```

Example of preprocessing using %TEMPFILE on a Windows or Unix system:

```
ncopy zospartner!"|ftscopy from,WindowsPC,(%tempfile),*any('hallo.txt'\
,trans='WindowTransadm')" – zosTransadm
```

openFT inserts the TSO or openFT commands specified as preprocessing in the FT request as instream data cards after the template statement "//SYSTSIN DD *". The TSO commands must comply with the IBM conventions. They can be of any length and, if necessary, openFT will spread the command over multiple lines. Any output from the openFT commands as part of preprocessing is redirected to SYSPRINT and consequently to the file referenced by RLFP. By default, TSO commands output to SYSTSPRT. It may be necessary to redirect this output to SYSPRINT and thus to the output file for preprocessing (e.g. LISTCAT OFILE(SYSPRINT)). When the preprocessing commands have been read in, openFT passes the subsequent commands to the internal reader for batch processing (this part is generated dynamically and cannot be modified).

```
//*****************************************************
//IFBAD IF (ABEND OR STEP2.RC>=12 OR NOT STEP2.RUN)
//          THEN
//STEP3     EXEC PGM=IKJEFT01
//OPENFT    DD DSN=&CONN##################################,
//          DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO##################################,
//          DISP=(SHR,KEEP)
//SYSPRINT  DD SYSOUT=*
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD *
FTDETP FAILURE &PRID
//          ELSE
//STEP4     EXEC PGM=IKJEFT01
//OPENFT    DD DSN=&CONN##################################,
//          DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO##################################,
//          DISP=(SHR,KEEP)
```

```
//SYSPRINT  DD SYSOUT=*
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD *
FTDETP SUCCESSFUL &PRID
//IFBADEND  ENDIF
/*
```

openFT then starts this job. If processing is aborted, openFT starts its own "Cancel-Job" on the basis of the job envelope TSOJOB. This job is assigned the letter "Z" as the last letter of the job name to give it a higher priority than the current processing jobs.

Example of the member TSOVFJOB

```
//&JOBP##N  JOB  &ACCN###################################,
//          MSGCLASS=X,
//          USER=&USID###,
//          NOTIFY=&USID###,
//          PASSWORD=&PASS###,
//          REGION=0M
//STEP0     EXEC PGM=IKJEFT01,
//          COND=(0,NE)
//OPENFT    DD DSN=&CONN#################################,
//             DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO#################################,
//             DISP=(SHR,KEEP)
//SYSTSPRT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSTSIN   DD *
FTATTP &PRID
//*****************************************************
//STEP1     EXEC PGM=IEFBR14
//STDOUT    DD DSN=&RLFP#################################,
//             DISP=(NEW,CATLG,DELETE),
//             DCB=(DSORG=PS,BLKSIZE=1536,RECFM=VB),
//             UNIT=SYSDA,SPACE=(1,(&PALC######,&SALC######)),
//             AVGREC=K
//STDERR    DD DSN=&RLFT#################################,
//             DISP=(NEW,CATLG,DELETE),
//             DCB=(DSORG=PS,BLKSIZE=1536,RECFM=VB,LRECL=1532),
//             UNIT=SYSDA,SPACE=(1,(256,2560)),AVGREC=K
//SYSERR    DD DSN=&RLFF#################################,
//             DISP=(NEW,CATLG,DELETE),
//             DCB=(DSORG=PS,BLKSIZE=1536,RECFM=VB,LRECL=1532),
//             UNIT=SYSDA,SPACE=(1,(256,2560)),AVGREC=K
//*****************************************************
//IFBAD     IF STEP0.RC=0 THEN
//STEP2     EXEC PGM=IKJEFT01,
//          COND=(0,NE)
```

```
//OPENFT    DD DSN=&CONN##################################,
//             DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO##################################,
//             DISP=(SHR,KEEP)
//SYSPRINT  DD DSN=*.STEP1.STDOUT,DISP=(MOD,KEEP,DELETE)
//SYSTSPRT  DD DSN=*.STEP1.STDERR,DISP=(MOD,KEEP,DELETE)
//SYSERR    DD DSN=*.STEP1.SYSERR,DISP=(MOD,KEEP,DELETE)
//SYSTSIN   DD *
```

A special form of preprocessing in z/OS takes the form of the server function for an "ftexec" command issued in partner system. ftexec expects to be returned the output from the passed commands (stdout), any error messages that occur (stderr) and an exit code.

openFT inserts the TSO or openFT commands specified as preprocessing in ftexec as instream data cards after the template statement "//SYSTSIN DD *". The TSO commands must comply with the IBM conventions. They can be of any length and, if necessary, openFT will spread the command over multiple lines. openFT dynamically appends the output from SYSTSPRT to that of SYSERR. To do this, it internally uses the IBM utility IEBGENER:

```
//IFBADEND  ENDIF
//STEP22    EXEC PGM=IEBGENER
//SYSUT1    DD DSN=*.STEP2.SYSTSPRT,DISP=(MOD,DELETE,DELETE)
//SYSUT2    DD DSN=*.STEP2.SYSERR,DISP=(MOD,KEEP,DELETE)
//SYSIN     DD DUMMY
//SYSPRINT  DD SYSOUT=*
//SYSTSIN   DD *
```

The content of the temporary file generated by this is redirected to "stderr" in the partner system and SYSPRINT is redirected to "stdout". After reading in the commands, openFT extends the job in accordance with the example for TSOVVJOB, see job steps STEP 3 and STEP 4 on page 79.

Example for the member TSONVJOB

```
//&JOBP##N  JOB  &ACCN###################################,
//          MSGCLASS=X,
//          CLASS=C,MSGLEVEL=(1,1),
//          USER=&USID###,
//          PASSWORD=&PASS###,
//          REGION=0M
//STEP1     EXEC PGM=IKJEFT01,
//          COND=(0,NE)
//OPENFT    DD DSN=&CONN########################################,
//             DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO########################################,
//             DISP=(SHR,KEEP)
//SYSTSPRT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSTSIN   DD *
FTATTP &PRID
//*****************************************************
//IFBAD     IF STEP1.RC=0 THEN
//STEP2     EXEC PGM=IKJEFT01,
//          COND=(0,NE)
//OPENFT    DD DSN=&CONN########################################,
//             DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO########################################,
//            DISP=(SHR,KEEP)
```

In the case of postprocessing in z/OS, the transferred data is first stored in a temporary file which is then available as input for the commands specified in the request (TSO commands or system commands for corresponding utilities). The temporary file can be directly referenced in the commands by means of the metastring %TEMPFILE.

openFT then extends the job as follows:

```
//SYSTSPRT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
//SYSTSIN   DD *
//IFBADEND  ENDIF
```

If %TEMPFILE is not used in the postprocessing commands, then postprocessing reads in the data via SYSUT1. The data stream from the FT request read via SYSTSIN then uses the data specified in SYSUT1 as input data. To this end, users can specify a corresponding utility program in the postprocessing operation in order to access this data.

openFT then extends the job as follows:

```
//SYSUT1    D     DSN=&RLFP##########################################,
//                  DISP=(MOD,KEEP,DELETE)
//SYSTSPRT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
//SYSTSIN   DD *
//IFBADEND  ENDIF
```

Example

The user wants to copy a file DAT1 from a Windows or Unix system to z/OS using ncopy. Via the postprocessing operation, the file DAT1 is to copy to another file DAT2 in the target system. The utility IEBGENER is used:

```
ncopy DAT1 ZosPartner!"|allocate dsname(DAT2) ddname(SYSUT2) mod keep
dsorg(ps) recfm(v,b) lrecl(259); 'call SYS1.LINLLIB(IEBGENER)'"
UserId,Account,Password
```

In all cases, the postprocessing job is extended by further statements after the processing commands in the same way as preprocessing jobs. This is performed in the same way as for the preprocessing job (see TSOVVJOB on page 78).

If processing is aborted, openFT starts its own "Cancel-Job" on the basis of the job envelope TSOJOB. This job is assigned the letter "Z" as the last letter of the job name to give it a higher priority than the current processing jobs, see TSOVVJOB with the job steps STEP 3 and STEP 4 on page 79.

2.7.2.4 Structure of the members SUCCMSG and FAILMSG

These members contain the installation-specific texts issued by openFT as a result of successful or unsuccessful file transfer to one or several consoles.

These members are evaluated only if a valid specification has been made for the SUCC_MSG or FAIL_MSG keyword in the FTMSPPAR member and if ENDMSG_ROUTCDE=1 has been set.

If the FTMSPPAR member contains valid specifications for SUCC_MSG or FAIL_MSG, and the members SUCCMSG and FAILMSG exist but are empty, no message is output.

The following rules apply to the installation-specific message texts:

- The text must begin and end with single quotes. These characters are not output but form part of the syntax.
- After replacement of the variables (see below), the text may be up to 102 characters long, **including** the single quotes in which it is enclosed.
- Like all other members of the FT parameter library, the members containing the message texts may not contain line numbering.

A single quote within a message no longer has to be duplicated as was the case in earlier versions. If you have already analyzed these messages prior to migration to V12, for instance using NetView, you should retain the duplicated quotes for reasons of compatibility.

Violation of these rules results in an **error**. At worst, either no message is output at all or a message is issued to **all** TSO terminals currently active.

Like the members PRTJOB, JCLJOB and TSOJOB, the message texts can contain variables which openFT replaces with the current values.

The following variables may be used in message texts:

FILX
: Name of the send or receive file as specified in the NCOPY command for the system involved. Maximum 58 characters in accordance with IBM conventions.

  When replacing this variable with the current value, openFT duplicates single quotes enclosing fully qualified file names, thus satisfying the syntax rule "If a single quote occurs in the message itself, it must be duplicated". (Single quotes are not duplicated in the message itself.) The maximum length of 58 characters for this parameter is formed as follows: 44 characters (maximum length of a fully qualified file name, not including the single quotes which enclose it) + 8 characters (maximum length of a member name) + 2 characters (parentheses enclosing the member name) + 4 (2 x 2 single quotes).

PNAM
: Symbolic name of the remote system. Maximum 8 characters in accordance with openFT conventions.

  If there is no symbolic partner name for this request, the first 8 characters of the partner address are output.

SUBM
: specifies the system in which the FT request was submitted. Maximum 6 characters. openFT replaces this variable with the following character strings:

  LOCAL if the request was submitted in the local system,

  REMOTE if the request was submitted in a remote system.

USID
: User ID from TRANSFER-ADMISSION.
  Maximum 7 characters in accordance with IBM conventions.

ACCX
: "accounting information" from the TRANSFER-ADMISSION.
  Maximum 42 characters in accordance with IBM conventions.

  When replacing this variable with the current value, openFT duplicates single quotes that can enclose "accounting information" (see section "Access authorization" in the manual "openFT (z/OS) - Command Interface"). This satisfies the syntax rule "If there is a quote, use double quotes" (only single quotes appear in the message itself).

  The maximum length of 42 characters for this parameter is formed as follows: 40 characters for the "accounting information" + 2 additional quotes.

OWID
: Owner of the FT request (user ID under which the FT request was submitted).
  Maximum 7 characters in accordance with IBM conventions. This variable is replaced with a valid value only for FT requests which were submitted in the local system. This variable is eliminated for FT requests submitted in a remote system.

PGRX
: The programmer's name as specified as a command prefix with the key PGRN= in the relevant command string for follow-up processing; i.e. in the member SUCCMSG, the value is taken from the command string in the parameter SUCCESS-PROCESSING, in the member FAILMSG it is taken from the command string in the parameter FAILURE-PROCESSING. For further details refer to the manual "openFT (z/OS) - Command Interface". Up to 20 characters, according to IBM conventions, plus the number of apostrophes possibly enclosing or contained in the current value.

When replacing the variable by the current value, openFT doubles the number of apostrophes that can enclose or be contained in a programmer's name. This fulfills the syntax rule "If an apostrophe occurs in a message, then double it". (Only single apostrophes appear in the message itself.)

This is also why the length of the field has to be increased by the number of apostrophes possibly enclosing or contained in the current value.

TRID
FT transfer identification. Maximum 10 characters (value range 1..2147483639) in accordance with openFT conventions.

The name of this variable must be given the prefix "%" or - for reasons of compatibility with predecessor versions - „&".

As many "#" fill characters as necessary should be appended to the names of these variables so that the field length reaches the maximum length (including the "&", for example &PNAM###). If openFT replaces the variables with the current values, it does not exceed the field length defined by the name of the variables including the fill characters; if necessary, the current values are truncated. Syntax errors caused by truncation can also lead to a **misfunction**.

In the opposite direction, openFT removes superfluous fill characters from current values that are shorter than these field lengths.

Example of the member SUCCMSG

(The FAILMSG member must have the same structure)

```
'DATASET␣&FILX####.....####␣TRANSFERRED␣
TO/FROM␣&PNAM###.'
```

The field &FILX####.....#### for the file name should have a total length of 58 characters. The first line of the member is 80 characters long (including the single quote at the start); the rest of the message (including the single quote at the end) is located on the second line. The total length of the message (maximum 98 characters including the single quotes) does not exceed the maximum permitted value of 102 characters.

If the NCOPY command was entered as follows:

```
NCOPY TRANS=TO,
      PARTNER=SYS1,
      LOC=(FILE='USER1.ABC',TRANS=(USER2,ACC2,PASS2)),
      REM=...
```

then the following message is issued in this example after successful file transfer:

```
DATASET 'USER1.ABC' TRANSFERRED TO/FROM SYS1. CN(00)
```

2.7.2.5 Structure of the member FNAMECTB

This element contains information on which file-specific character sets openFT is to use and on which files are to be encoded with which character sets. A range of character sets in the form of code tables are supplied with openFT. See also section “Administering code tables” on page 125.

The character set is selected by means of the name of the send or receive file. If openFT-AC is used, the file name may consist of the specification from the transfer request and from the admission profile accessed in the transfer request.

Each line of the member must contain one of the following specifications:

- the name of the character set (code table) in the following format:

  ```
  @ctabname          [comment]
  ```

  ctabname is the name of the code table (1 to 8 characters, also known as the CCS name).

- a selection pattern for selecting file names where the "*" character can be used as a placeholder for a part of the file name; the "*" character may only be used once in a selection pattern. It stands for an optional number of characters (including 0 characters).

The following rules must be observed when creating the member FNAMECTB:

- There must be no blank between "@" and the name of the character set.
- A line containing the name of a character set may be followed by one or more lines with selection patterns for file names. All files whose names match one of the selection patterns are allocated to that character set.
- A table name to which no selection pattern is allocated is skipped.
- If a table name is specified several times, the character set is loaded several times; all specified allocations to selection patterns are considered.
- If a selection pattern is specified more than once, the first allocation applies.
- If a file name matches several selection patterns, the first hit applies. More specific selection patterns must therefore be located before a more general selection pattern.
- Leading blanks in a line are ignored so that the allocation specifications can be clearly structured.
- All lines up until the first occurrence of a table name are treated as comment lines.
- At the end of each line, a comment can be included, separated by at least one blank from the rest of the line.

- File name patterns not enclosed in single quotes ignore the first-level qualifier if the files are not openEdition files.

  Example for the pattern *A.TEXT:

  'USERA.TEXT' does not match, because the A is part of the first-level qualifier

  'USER1.AAA.TEXT' matches

  '/AAA.TEXT' matches, because it is an openEdition file

openFT assigns a table ID to each code table that is loaded. A list of all code tables is entered into the openFT job log; the table IDs and the allocated selection patterns for file names are also listed there for each code table. Incorrect specifications in FNAMECTB and errors that occur when a table is loaded are indicated by means of a negative number as table ID. The individual values have the following meanings:

Table ID	**Meaning**
-1	Syntax error in the selection pattern specification
-2	Code table could not be loaded
-3	Syntax error in the specification of the name of a code table, for instance a blank between the "@" and the name of the code table or the name is too long (in this case the first 8 characters of the name are shown)

Errors that occur when the member FNAMECTB is read in are not logged; the allocation list that was created up until the first error occurred is used.

Example for the member FNAMECTB

In this example, the possible entries (including some erroneous entries) in the member FNAMECTB are presented along with effect and including the entries in the openFT job log which they generate.

```
THIS MEMBER DEFINES FILE SPECIFIC CODE TABLES FOR OPENFT

@FNCOD001         - SPECIAL TABLE 1
   'USER1.*.TEXT'
@FNCOD002         - SPECIAL TABLE 2
   'USER2.ABC*'
   'USER2.*ABC'
   *TEXT*
   'USER3.DEF'
@IBM037         - EBCDIC CODE PAGE 037
   *.TEXT
@IBM273          - EBCDIC CODE PAGE 273
   DEF
@IBM500          - EBCDIC CODE PAGE 500
   *.CHAR
@FNCODTABL
   XYZ*
@ FNCOD000
   *CHAR
@UTF8
   *.tst
```

The resulting entries in the openFT job log:

```
TABLE_ID CODETABLE PATTERN
0014FE00 FNCOD001  'USER1.*.TEXT'                          > (1)
0014FC00 FNCOD002  'USER2.ABC*'                            > (2)
0014FC00 FNCOD002  'USER2.*ABC'                            > (3)
-1       FNCOD002  *TEXT*                                  > (4)
0014FC00 FNCOD002  'USER3.DEF'                             > (5)
0014FA00 IBM037    *.TEXT                                  > (6)
-2       IBM273    DEF                                     > (7)
0014F800 IBM500    *.CHAR                                  > (8)
-3       FNCODTAB  XYZ*                                    > (9)
-3       FNCOD00   *CHAR                                   > (10)
001E445F  UTF8     *.tst                                   > (11)
```

Explanations

(1) All files with the first level qualifier USER1 whose names end with ".TEXT" are coded using the character set FNCOD001.

(2) All files with the first level qualifier USER2 whose partially qualified names begin with "ABC" are coded using the character set FNCOD002.

(3) All files with the first level qualifier USER2 whose names end with "ABC" are also coded using the character set FNCOD002.

(4) Syntax errors in the specified selection patterns: the "*" character is used several times.

(5) The file 'USER3.DEF' is also coded using the character set FNCOD002.

(6) All files whose names end with ".TEXT" are coded using the character set IBM037 (except those whose first level qualifier is USER1, see (1)).

(7) All files with a partially qualified name DEF (except 'USER3.DEF' - see (5)) are to be are coded using the character set IBM273, but this table cannot be loaded, for instance because it has been deleted from the library.

(8) All files whose names end with ".CHAR" are coded using the character set IBM500.

(9) Syntax errors in the specification of the name of the code table: Name longer than 8 characters. Note: This entry is to capture all files whose partially qualified name begins with "ABC" but with the exception of the files whose first level qualifier is USER2 - see (2).

(10) Syntax errors in the specification of the name of the code table: There is a blank between "@" and the name of the code table. - Note: This entry is to capture all files whose names end with "CHAR" with the exception of the files whose names end with ".CHAR" - see (8).

(11) All files with names ending in ".tst" are encoded using the UTF8 character set.

2.7.2.6 Structure of the member FTACPAR

The installation parameters which are needed when openFT-AC is used are stored in this member.

When the openFT load module is started for the first time after the delivery unit openFT-AC is installed, openFT automatically generates the FTAC file (see section “Internal openFT data sets” on page 193) using the characteristics specified for its name and size stored in this member or the default values.

Each line of the FTACPAR member can contain exactly one parameter in the form "keyword=value". No blanks may be inserted between "keyword", "=" and "value". Below is a list of the keywords which may be used.

Keywords:

FILE_2ND_Q=
: The second level qualifier for the name of the components of the FTAC file (see section “Internal openFT data sets” on page 193).
Up to 17 characters (default: <inst>.SYSFSA, where <inst> is the name of the openFT instance). For the sake of clarity, the name should always start with the instance name followed by a period.

FILE_SIZE_KB=
: Initial size of the FTAC file (in KB).
Maximum value: 30736382 (default: 1024).

  openFT uses this value as the primary allocation size when creating the VSAM cluster which is part of the FTAC file. The value is halved for the size of the secondary allocation.

  The specified maximum value is the program-technical limit. When choosing a value for the initial size of the FTAC file, the actually available storage space needs to be taken into consideration. Note that the FTAC file is created on the same data volume as the request file, the partner list, the operating parameter file and the logging file (see section “Internal openFT data sets” on page 193).

Example for the FTACPAR member

```
FILE_2ND_Q=HAPPI.OPENFTAC
```

2.7.3 Providing the OPFT subsystem

The commands in the dialog tasks with which FT users and administrators work (NCOPY or an alias, or FTHELP or FTTRACE) are encrypted for the purposes of internal communication with openFT. This encryption (and decryption) is performed by the OPFT subsystem. In addition, OPFT administers the running openFT instances. Optionally, it is also possible to use the MSG_CRYPT parameter in the PARM library to activate message encryption. OPFT must be installed in the computer's IPL. To do this, it is necessary to copy the members from OPENFT.LPALIB to SYS1.LPALIB or store them in a user LPALIB that is concatenated with this library.

- IGX00211 (SVC handler)
- OPFTIGX
- OPFTINIT (Startup routine of the OPFT subsystem)
- OPFTSUB (Subsystem handler)

It is important to avoid name conflicts with load modules that already exist in SYS1.LPALIB.

The portal to the subsystem is implemented via SVC 109 with "extended code 211". If "extended code 211" is already used for a different purpose in your system then the samples in openFT provide a procedure with the name LINKIGX that allows you to generate other, alternative "extended codes" that can be set in the PARM file with 'OPENFT_SVC='.

The subsystem is initialized the first time openFT is started after IPL. The start of the subsystem is confirmed by a console message. After initialization, the subsystem remains active until the next IPL. No further administration is necessary and, in particular, the subsystem does not require any start parameters.

2.7.4 openFT as a job or started task

The OPENFT load module runs either as an ordinary batch job or as a started task. In both cases, the associated user ID must possess the authorizations described in the section "openFT privileges" on page 38.

The FJGEN command (see page 56) generates JCL statements for loading a batch job and starting the openFT load module. This JCL is entered in the FJBATCH member of the FT procedure library <openft qualifier>.<inst>.CLIST.

Example of the FJBATCH member

```
//OPENFTF JOB (A123,B123),
//          CLASS=A,MSGCLASS=A,
//          USER=OPENFT,PASSWORD=OPENFT,
//          TIME=1440,REGION=0M
//DLTDMP    EXEC PGM=IEFBR14
//DELFILE   DD DSN=OPENFTQU.STD.SYSUDUMP.PREV,
//             DISP=(MOD,DELETE,DELETE),
//             SPACE=(CYL,(20,5)),
//             DCB=(DSORG=PS)
//RENAME    EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//SYSTSIN   DD *
//SYSIN     DD *
//             ALTER 'OPENFTQU.STD.SYSUDUMP' +
//             NEWNAME ('OPENFTQU.STD.SYSUDUMP.PREV')
//             IF LASTCC = 8 THEN SET MAXCC = 0
//OPENFT    EXEC PGM=OPENFT,TIME=1440,
//          PARM='OPENFTQU,VSN123/SYSDA,A,FTID1,STD,AFFE,1100,'
//*            openFT V12.1A00 / FJBATCH V121A00
//STEPLIB   DD DSNAME=OPENFT.OPENFT.LOAD,
//             DISP=(SHR,KEEP)
//OPENFTS   DD DSNAME=OPENFT.OPENFT.NCLOAD,
//             DISP=(SHR,KEEP)
//OPENFT    DD DSNAME=OPENFTQU.STD.CONN,
//             DISP=(SHR,KEEP)
//OPFTATT   DD DSNAME=OPENFTQU.STD.OPFTATT,
//             DISP=(SHR,KEEP)
//*DDUADS    DD DSNAME=SYS1.UADS,
//*             DISP=(SHR,KEEP)
//OPFTHSM   DD DSNAME=OPENFTQU.STD.COLLECT.DATA,
//             DISP=(SHR,KEEP)
//MCDS      DD DSNAME=DFHSM.MCDS,DISP=SHR
//SYSIN     DD DUMMY
//SYSOUT    DD DUMMY
//IEBCOUT   DD DUMMY
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD DSN=OPENFTQU.STD.SYSUDUMP,
//             SPACE=(CYL,(20,5)),DISP=(,CATLG),
//             DCB=(DSORG=PS)
```

The JCL card for SYS1.UADS has been commented out as it is not required when RACF is used.

The DD cards OPFTHSM and MCDS are required for archiving and retrieving files.

FJGEN creates the FJBATCH member using the specified installation parameters (for further information, see the description of the FJGEN command, page 104).

You can adapt the JCL statements in this procedure to meet the requirements of your installation.

For example, if a file named SYS1.UADS exists on your system whereas validation of the user ID is to be performed via RACF then you must remove the two lines with the assignment of DDUADS to SYS1.UADS from the procedure.

In the statement `//OPENFT  EXEC PGM=OPENFT, ...` you can replace the program name OPENFT with the aliases OPENFTS or OPENFTSL described in section “Installing from CD” on page 42, see also page 43.

If a local host name other than the default has been specified for the openFT instance in FJGEN then this is entered after the port number in the PARM parameter of the //OPENFT ... statement.

You can redirect the openFT job log to a file by modifying the DD statement with the label `SYSPRINT`. Attention must be paid to the following factors:

– Output can be directed into a PS data set or into a PO or PDSE member. However, an existing PO or PDSE member cannot be extended.

– If the file is to be newly created, do not make any specifications concerning the record length (LRECL) and the block size (BLKSIZE). openFT generates the file with LRECL=1536 and BLKSIZE=1536.

– If you are using an existing file (i.e. if you are extending a PS data set or if you are creating a new member in an existing PO or PDSE data set), the file must have the attributes LRECL=512 and BLKSIZE=512.

Examples

– If the PS data set does not yet exist, it is to be created; otherwise it is to be extended:

```
//SYSPRINT DD DSN=USERID.LOG1,DISP=(MOD,CATLG),RECFM=FB,
//         SPACE=(CYL,(20,20))
```

– An existing PS data set is to be overwritten:

```
//SYSPRINT DD DSN=USERID.LOG2,DISP=(OLD)
```

– The PO or PDSE data set already exists. If the member does not yet exist, it is to be created; otherwise it is to be overwritten:

```
//SYSPRINT DD DSN=USERID.LOG3(MEMBER1),DISP=(OLD)
```

The DD statement with the label `SYSUDUMP` causes openFT to write the dump to this file in printable form on a "Cancel with Dump". Other system dumps are output to SYSFDF.

If openFT is to run as a started task, which means that it is to be started either automatically when the system starts or by means of an operator command, a specific start procedure must be created by the user's computer center. The FJBATCH created with the FJGEN can be used as a template to be copied.

Example of a start procedure:

```
//          PROC
//DLTDMP    EXEC PGM=IEFBR14
//DELFILE   DD DSN=OPENFTQU.STD.SYSUDUMP.PREV,
//             DISP=(MOD,DELETE,DELETE),
//             SPACE=(CYL,(20,5)),
//             DCB=(DSORG=PS)
//RENAME    EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//SYSTSIN   DD *
//SYSIN     DD *
//             ALTER 'OPENFTQU.STD.SYSUDUMP' +
//             NEWNAME ('OPENFTQU.STD.SYSUDUMP.PREV')
//             IF LASTCC = 8 THEN SET MAXCC = 0
//OPENFT    EXEC PGM=OPENFT,TIME=1440,
//          PARM='OPENFTQU,VSN123/SYSDA,A,FTID1,STD,AFFE,1100,'
//*           openFT V12.1A00 / FJBATCH V121A00
//STEPLIB   DD DSNAME=OPENFT.OPENFT.LOAD,
//             DISP=(SHR,KEEP)
//OPENFTS   DD DSNAME=OPENFT.OPENFT.NCLOAD,
//             DISP=(SHR,KEEP)
//OPENFT    DD DSNAME=OPENFTQU.STD.CONN,
//             DISP=(SHR,KEEP)
//OPFTATT   DD DSNAME=OPENFTQU.STD.OPFTATT,
//             DISP=(SHR,KEEP)
//OPFTHSM   DD DSNAME=OPENFTQU.STD.COLLECT.DATA,
//             DISP=(SHR,KEEP)
//MCDS      DD DSNAME=DFHSM.MCDS,DISP=SHR
//SYSIN     DD DUMMY
//SYSOUT    DD DUMMY
//IEBCOUT   DD DUMMY
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD DSN=OPENFTQU.STD.SYSUDUMP,
//             SPACE=(CYL,(20,5)),DISP=(,CATLG),
//             DCB=(DSORG=PS)
```

The installation parameters described in the section the FJGEN command (see page 104) must also be used in this start procedure.

The explanations given above for adapting the FJBATCH member (program name OPENFT, SYSPRINT, SYSUDUMP) also apply here.

2.7.5 Loading and starting the openFT load module

By using the FJGEN command and the entries associated with it a batch job was created in the openFT instance's FT procedure library in the member FJBATCH. This job can now be executed with the FJINIT command.

The member FJBATCH can be adapted to the conventions of your computer center if the batch job does not comply with them.

Alternatively, openFT can also be started as a started task.

For more details, see the description of the commands FJGEN (page 104) and FJINIT (page 115) as well as the section "openFT as a job or started task" on page 92.

2.7.6 Activating, deactivating and terminating openFT

After the openFT load module has been loaded, openFT can be activated using the FTSTART command.

File transfer requests cannot be accepted or executed until openFT has been activated.

openFT is deactivated using the FTSTOP command. Once this command has been issued, no more file transfer requests can be executed. openFT is terminated using the FTTERM command. If necessary, this command deactivates openFT (FTSTOP command) and terminates the openFT job.

2.8 Linking openFT with data protection products

For each file transfer request and file management request, openFT checks

– the user's access authorization to the system (transfer admission)
– the user's access authorization to the relevant file
– if preprocessing, postprocessing or follow-up processing is to be triggered following a file transfer request: the user's authorization to do so.

Users must demonstrate their authorization by means of the specifications they make in the TRANSFER-ADMISSION and PROCESSING-ADMISSION operands for the system involved. Transfer requests in which authorization is not demonstrated satisfactorily are rejected.

If FTAC is not used, the user must make the entries required for checking his transfer admission directly in TRANSFER-ADMISSION or PROCESSING-ADMISSION (i.e. LOGON ID consisting of user ID, account number and password). If FTAC is used, a TRANSFER-ADMISSION defined in an admission profile can be specified instead of the LOGON ID. FTAC will then read the information needed for the admission check from the relevant profile (i.e. the LOGON ID consisting of user ID, account number and password).

openFT checks the user's transfer admission using RACF calls or against the entries in the MVS system file SYS1.UADS. Transfer admission can also be checked using RACF calls or by calling the PROTECT macro (for more information, see below). To this end, openFT must be assigned APF authorization (see the section “openFT privileges” on page 38) or read access to SYS1.UADS. openFT does not have write access to SYS1.UADS or to RACF lists.

Since all RACF calls are handled by the RACROUTE macro, it is possible to connect an installation-specific MVS exit routine to the **MVS Router exit** or to use an RACF-compatible software product such as ACF-2 or TOP-SECRET (If TOP-SECRET is used, openFT identifies itself to TOP-SECRET as "OSFSUBT", i.e. "PGM=OSF" must be specified).

Information on the requirements which must be met by an RACF-compatible software product in order to enable openFT to perform system and data access control via this product is given in the product-specific manuals.

The interface of the MVS Router exit is described in the IBM manual "System Programming Library: Resource Access Control Facility (RACF)".

openFT accesses the file SYS1.UADS via the DD name DDUADS (see the corresponding DD statements in the examples in the section “openFT as a job or started task” on page 92). openFT checks whether the file SYS1.UADS is available; this check is carried out only during processing of the first transfer request after loading and starting the openFT load module. If this file is not available (DD statement is missing, file is not available or not

readable, etc.), openFT no longer accesses the SYS1.UADS file until termination of the openFT load module. During the processing of all subsequent transfer requests, the SYS1.UADS file is considered to be unavailable.

Notes

- If the transfer request is rejected during synchronous command processing, the NCOPY command is terminated with the return code X'0C'. This also applies to the NCOPY program interface.
- Whether or not follow-up processing takes place following rejection of a transfer request (FAILURE-PROCESSING) depends on which FT system rejects the transfer request:
  - If the transfer request is rejected by the **local** openFT instance, no follow-up processing takes place in either of the two FT systems involved.
  - If the transfer request is rejected in the **remote** system, no follow-up processing takes place in the remote system. In this case, the follow-up processing for unsuccessful file transfer (FAILURE-PROCESSING) is initiated in the local system.

  The message that is issued (e.g. FTR2047, FTR2169) indicates whether the local or remote system rejected the transfer request.

2.8.1 Checking the transfer admission

When FTAC is used, the transfer admission check for file transfer and file management requests is carried out when the request is checked by FTAC, independent of whether the request contains a LOGON ID (user ID, account number, password) or a TRANSFER-ADMISSION defined in a user profile. In the latter case, FTAC reads the information required for the check (i.e. the LOGON ID consisting of user ID, account number, password) from the relevant profile. Like openFT, FTAC performs the transfer admission check using RACF calls or compares the entries with the information contained in the SYS1.UADS system file.

If FTAC is not used, openFT itself checks the transfer admission using the LOGON ID (user ID, account number, password) contained in the request.

The user must provide evidence of his or her transfer admission in TRANSFER-ADMISSION. The check sequence for transfer admission (TRANSFER-ADMISSION) is the same as for checking the admission for follow-up processing (PROCESSING-ADMISSION, see page 102); so both cases are treated the same.

Checking the user ID and password

openFT first uses the RACROUTE macro to check whether the user ID specified in the NCOPY command for the TRANSFER-ADMISSION or the PROCESSING-ADMISSION is valid and whether the associated user password, if any, is correct. (The RACROUTE macro makes use of the RACF macro RACINIT for this purpose.) If the result of this check is negative, the transfer request is rejected and an error message is issued.

If the return code from the RACROUTE macro indicates that neither RACF nor a compatible product (ACF-2, TOP-SECRET) is installed and active, openFT attempts to check the user ID and the associated password using the SYS1.UADS data set.

If the SYS1.UADS data set is also unavailable, no transfer request is processed and an error message is issued.

Checking the account number

openFT takes the account number from the user's specification in the NCOPY parameter ACCOUNT (TRANSFER-ADMISSION or PROCESSING-ADMISSION). If "accounting information" is specified here, openFT extracts the account number from this information. Any "(" and/or " ' " characters at the start of this specification are removed. The string is then searched for the first comma ",". If a comma is found, all characters **preceding** this comma are interpreted as the account number. If, however, one of the characters ")" or " ' " is found first, all characters **preceding** this character are interpreted as the account number. If none of the characters "," or ")" or " ' " are found, the entire string is interpreted as the account number.

If the SYS1.UADS data set is available, openFT checks the account number against the entry which matches to the user ID in this file. If the account number is not entered here, the transfer request is rejected and an error message is issued. If the SYS1.UADS data set is available and no account number is specified in the corresponding operand of the NCOPY command (TRANSFER-ADMISSION or PROCESSING-ADMISSION), **no** check is performed on the account number.

If the SYS1.UADS data set is not available, openFT checks whether RACF (or compatible product) is active and whether the RACF resource class ACCTNUM is active. If this is the case, openFT checks the account number using RACF. If an account number is specified in the corresponding operand of the NCOPY command (TRANSFER-ADMISSION or PROCESSING-ADMISSION), this is used for checking purposes. However, if no account number is specified, openFT looks for the "TSO default account number" in the "TSO segment" (see page 100) of the user-specific data relating to the user ID in the ACF database. (The RACROUTE macro uses the RACF macro RACXRTR for this purpose). If this database contains a value with a maximum length of 40 characters, it is used for checking purposes. However, if it is still not possible to find an account number specification, a "pseudo account number" consisting of 40 "@" characters is used.

The RACROUTE macro uses the RACF macro RACHECK to perform an RACF check of the account number.

If the SYS1.UADS data set is not available and RACF is active, but the RACF resource class ACCTNUM is not active, no default account number is allocated and no account number check is performed.

If the SYS1.UADS data set is not available and RACF is not active, then the transfer request was rejected with an error message when the user ID and password were checked (see above).

A computer center can thus ensure that transfer requests are processed without the user having to specify an account number in the NCOPY command. To this end, the following steps must be taken:

- SYS1.UADS data set available: no further steps necessary.
- SYS1.UADS data set not available, RACF resource class ACCTNUM inactive no further steps necessary
- SYS1.UADS data set not available, RACF resource class ACCTNUM active:

  In this case there are two possibilities:

  – If omission of the account number will cause openFT to use the default account number of the user ID specified in TRANSFER-ADMISSION or PROCESSING-ADMISSION, then no further steps are necessary. The RACF database, must contain a default account number for each relevant user ID.

  – If no default account numbers are used and you do not want openFT to check the account numbers of selected users, you must ensure that the above-mentioned "pseudo account number" (40 "@" characters) is entered in RACF (resource class ACCTNUM), and that only these selected users are authorized to use this "pseudo account number". These users may not then make any specification in the ACCOUNT parameter of the NCOPY command.

Notes on the TSO segment

If file transfer requests are initiated from a user ID that does not have a TSO segment or for which there is no standard account information then the local TRANSFER-ADMISSION together with the user ID and account (without user password) must be specified in the NCOPY/FTACOPY/FTSCOPY command. It is not possible to call the FTEXEC and FTADM commands from IDs without a TSO segment. If this restriction is not respected then the request is rejected with the message FTR2047.

2.8.2 Checking access authorization

Access authorization for the file accessed by the request is always performed by openFT itself, regardless of whether FTAC is used. The access authorization is checked after a positive transfer admission check (see previous section). The access authorization is checked for the user named in the request's TRANSFER-ADMISSION or for the user determined by FTAC using the information in the TRANSFER-ADMISSION (in the relevant profile) or by the openFT-specific exit routine

The procedure for checking access authorization distinguishes between read access (send file) and write access (receive file). If the user does not have the appropriate access authorization, the transfer request is rejected.

A distinction must be made between the following cases:

- RACF is installed and active:

  openFT uses RACF to check the user's authorization to access the send or receive file (read or write access). The RACROUTE macro makes use of the RACF macro RACHECK with the resource class DATASET for this purpose. For technical reasons the RACROUTE macro again calls on the RACF macro RACINIT to supply the user ID specified in TRANSFER-ADMISSION, generally together with the associated user password.

- RACF is not installed or is not active:

  In this case, the user's authorization to access a file is checked only in the case of a receive file which is password-protected according to the catalog entry. The file password specified in the NCOPY command for the receive file is then checked against the relevant entry in the PASSWORD file of the system (PROTECT macro). No password check takes place for send files (which can only be read).

2.8.3 Checking authorization for follow-up processing

Users must demonstrate their authorization to initiate follow-up processing by specifying a PROCESSING-ADMISSION. If the PROCESSING-ADMISSION is not explicitly specified, the data are taken from the TRANSFER-ADMISSION for the PROCESSING-ADMISSION. In the case of follow-up processing, the parameters USER-ID, ACCOUNT and PASSWORD must be explicitly assigned a value in one of the two ADMISSIONs. When FTAC is used, the data for the PROCESSING-ADMISSION can also be specified in an admission profile.

The authorization to initiate follow-up processing is checked by openFT.

The description given in "section “Checking the transfer admission” on page 98" applies when the authorization to initiate follow-up processing is checked by openFT.

A special case of follow-up processing under openFT is the character string "ALLOC DSNAME(...)". This special openFT statement is used to specify the name of a cataloged PS data set or the name of a member of a cataloged PO or PDSE data set containing a complete executable job.

openFT starts this job within follow-up processing via the Internal Reader. In this case, openFT does **not** generate any additional job control statements. In this way, it is possible to execute follow-up processing jobs with user-specific job parameters.

In this special case:

- openFT checks the access authorization of the user to this file on the basis of the data in the PROCESSING-ADMISSION. This transfer admission check is always performed after successful checking of the authorization for initiating follow-up processing.
- However, openFT does **not** check the values for user ID, account number and password specified in this PS data set or in this member. In order to prevent users from accessing the system unchecked, you are therefore recommended to use openFT-AC.

2.8.4 Checking preprocessing and postprocessing authorizations

The authorization to perform preprocessing and postprocessing corresponds to the admission under which the file transfer itself was performed (see "Checking access authorization"). It is therefore not taken from the PROCESSING-ADMISSION. The admission for z/OS is either proven explicitly on the basis of the USER-ID, ACCOUNT and PASSWORD specifications in the local TRANSFER-ADMISSION or implicitly through the use of an admission profile. In an admission profile that is to be used for preprocessing or postprocessing, the USER-ID, ACCOUNT and PASSWORD specifications must be stored in the USER-ADMISSION.

2.9 Configuring FTAC

Authorization of the FTAC administrator

It is recommended that the authorization to administer FTAC be given to those users in the system who are responsible for data protection in an z/OS system, since they are the best placed to know what protection measures are required where.

The FTAC administrators of an openFT instance are defined in the FTACADM member of the PARM parameter library (see page 70).

Adapting the default admission set

After the installation of FTAC, all values of the default admission set are **set at 0**!

This means that it is not yet possible to execute a file transfer with the local openFT instance. This is because as long as no other admission sets are made with FTMODADS, the default admission set is valid for all user IDs. The maximum security level 0 for the basic functions means that these basic functions may not be used. An FTAC administrator must therefore use the command FTMODADS to raise the values of the default admission set.

Examples

1. All partner systems should be accessible for file transfer for all FTAC users. This is achieved by setting all the values of the default admission set to100. The following command is used:

   ```
   FTMODADS␣*STD,MAX-LEV=100
   ```

   More information on the command FTMODADS can be found in the manual "openFT (z/OS) - Command Interface".

2. A differentiated setting of the default admission set might look as follows:

   ```
   FTMODADS USER-IDENTIFICATION=*STD,                                  -
            MAX-LEVELS=(OUTBOUND-SEND=50,OUTBOUND-RECEIVE=50,         -
                        INBOUND-SEND=20,INBOUND-RECEIVE=20,           -
                        INBOUND-PROCESSING=10,INBOUND-MANAGEMENT=0)
   ```

   The different security levels are assigned selectively. For example, the function "inbound management" can be fully blocked by setting the security level to 0.

WARNING!
Note that FTAC is only effective for connected products such as openFT. If other file transfer products without an openFT-AC connection are also being used, a more comprehensive and coordinated security concept would be advisable.

A key pair set must be created with FTCREKEY to be able to use the Crypto module.

2.10 openFT commands for installation and initial operation

For installation and initial operation, openFT provides the following commands:

FJGEN	Setting installation parameters
FJGENPAR	Output installation parameters
FJINIT	Load openFT
FTUPDPAR	Update operating parameters

2.10.1 FJGEN - Set installation parameters

Note on usage

User group: FT administrator

Functional description

You use the TSO procedure FJGEN to set up a new openFT instance or to modify the parameter settings of existing instances. The FJGEN command can only be issued in TSO command mode:

```
EXEC <FT-basic-procedure-library>(FJGEN)
```

where <FT-basic-procedure-library> must be replaced by the CLIST present in the FJGEN command (generally OPENFT.CLIST under the openFT installation ID).

FJGEN starts a dialog that requests the installation parameters for the openFT instance. Filenames must be entered with the user ID but without single quotes.

FJGEN uses the installation parameters to create installation-specific CLISTs and the JCL for an installation-specific batch job (see below). These procedures are required for the administration of openFT. FJGEN stores them in the FT procedure library (CLIST library):

```
<openft qualifier>.<inst>.CLIST
```

The first two name parts here are replaced by OPENFT QUALIFIER and INSTANCE NAME.

FJGEN can also be used without an operand to modify the installation parameters; the procedures mentioned are then regenerated. The changes take effect the next time the installation-specific batch job is started with FJINIT.

The batch job and the FJINIT command are located in the CLIST generated by FJGEN: <openft qualifier>.<inst>.CLIST.

Even if openFT is running as a started task, the installation parameters are modified with FJGEN. The FT administrator must make the necessary changes in the start procedure himself/herself; see section “openFT as a job or started task” on page 92. In the case of parameters that are queried by FJGEN but are not required for the started task, the best solution is to enter an "x" in FJGEN.

Note

You can also store some of the installation parameters in the PARM member of the FT parameter library and pass them to openFT; specifications of this kind overwrite the specifications made for FJGEN. Further information is given in the section “Setting up the FT parameter library” on page 57.

Example: Set installation parameters (FJGEN without an operand)

```
fjgen
 *****  FJGEN/V121A00 INSTALLATION PROCEDURE openFT V12.1A00 *****
 ENTER INSTANCE NAME       : (DEFAULT: STD)
 ENTER FT-LOADLIB          : USERA.openft.load
 ENTER FT-NCLOADLIB        : USERA.openft.NCLOAD
 ENTER VOLUME/UNIT         : vsn123/sysda
 ENTER openFT USER ID      : openft
 ENTER openFT USER ACCOUNT : (a123,b123)
 ENTER openFT USER PASSWORD: openft
 ENTER OPENFT QUALIFIER    : openftqu
 ENTER FT-ID               : ftid1
 ENTER FT-PASSWORD         : affe
 ENTER RUNMODE             : S(TANDARD)/A(UTOMATIC)
 ENTER FT-PARMLIB          : openftqu.std.parm
 ENTER CMDPORT             : 1100
 ENTER HOST NAME           :
 ENTER HSM-MCDS NAME       :

 FJGENPAR CREATED
 FJINIT   CREATED
 FJBATCH  CREATED
 FJVERS   CREATED (FUNCTION: GET VERSION OF LOADMODULS)
 ******   FJGEN END   ******
 READY
```

The FT procedure library FT-PROCLIB is additionally displayed on the subsequent call to FJGEN 'INFO' or FJGENPAR; see page 106.

The various items of information requested or displayed have the following meaning:

INSTANCE NAME
The instance name is used to administer the openFT instance. It may be up to 5 characters in length. If this entry is omitted then the instance name STD is set.
The names of instances that are to be switched within a computer cluster must be unique within the cluster.

The instance name identifies the components that belong to an openFT instance (data sets) and is used to address these internally (see also section “Setting up openFT instances” on page 156). However, it should not be confused with the instance identifier (this is defined for the purposes of address information with the FTMODOPT command).

FT-PROCLIB
Name of the FT procedure library (CLIST library). This is only displayed with FJGEN ’INFO’ or FJGENPAR (see page 113).
If it does not already exist, this PO file is created automatically when FJGEN is called. FJGEN stores the command procedures for the openFT instance in this library. It has the fixed name <openft qualifier>.<inst>.CLIST
The first two name parts here are replaced by OPENFT QUALIFIER and the instance name.

FT-LOADLIB
Name of the FT load module library. This PO or PDSE data set must contain the following load modules: OPENFT, OPENFTSL and OPFTSUBL. The name of the library must be entered including its user ID but without single quotes.

FT-NCLOADLIB
Name of the FT load module library for openFT commands such as FTSHWPTN, NCOPY, etc. Among other things, this PO or PDSE file must contain the load modules FTATTP and FTDETP. The name of this library must be entered with the user ID but without quotes.

VOLUME/UNIT
VSN (volume serial number) and group name (unit) of the disk containing the request file, the partner list, the log file, the FTAC file, the trace files and the dump files, if any (see section “Internal openFT data sets” on page 193).

If the corresponding files are SMS managed, the specifications for VOLUME and UNIT may have no effect under certain circumstances. If the files are not SMS managed, an "SMS managed volume" must not be specified here.

You can specify both values; if you only want to specify one of the two values, it may be necessary to use a slash to distinguish which value you want to specify. If you want to specify neither VOLUME nor UNIT (i.e. only a slash), openFT assumes the UNIT name DASD. This UNIT name must therefore be defined in the system. (You can also define the volume for the request file, the partner list and the volume for the trace files and dump files via the corresponding parameters in the PARM member. Specifications in PARM overwrite the specifications made for FJGEN. Further information is provided in the section “Setting up the FT parameter library” on page 57. The assumption is made here that no volume

specifications are made in PARM.)

Examples

VOLUME/UNIT	VOLUME	UNIT
VSN123/SYSDA	VSN123	SYSDA
VSN123 or VSN123/	VSN123	---
/SYSDA	---	SYSDA
/	---	DASD

openFT USER ID
User ID under which the openFT job is to execute. Once an instance has been set up, this user ID is also authorized by default to administer FT and possibly also FTAC.

openFT USER ACCOUNT
Accounting information for the job under which openFT is to execute. If the accounting information contains more than one parameter it must be specified in parentheses (see IBM manual "MVS/ESA JES2 Commands").

Null input is permissible if no accounting information is required.
Maximum length of accounting information: 40 characters.

openFT USER PASSWORD
Password for the user ID under which openFT is to execute.

OPENFT QUALIFIER
Qualifier for the instance-specific files. The OPENFT QUALIFIER may be up to 17 characters in length and may contain maximal a period. Hence, It may consist solely of a "first level qualifier" or a "first level qualifier" and a "second level qualifier".

Please note the following:
– The "second level qualifier" in the OPENFT QUALIFIER may consist of at most one character if ADM traps are to be output.
– Trace file names can be shortened if the OPENFT QUALIFIER contains a "second level qualifier".
– The "switch log files" function in the FTMODOPT command works only to a limited degree if the qualifier is longer than 11 characters. If a "second level qualifier" is defined (LOGFILE_2ND_Q parameter in the parameter library, PARM member), there are restrictions if the two qualifiers together are more than 23 characters in length.

FT-ID
FT identifier. This character string can consist of up to 5 alphanumeric characters and must be unique among all FT systems interconnected via an SNA network. If an SNA network is not used either for internal communication or for interconnections with other FT systems, then you can specify any value for FT-ID (preferably an 'x').

FT-PASSWORD
FT password. This password serves to protect the VTAM applications, the request file, the partner list and the trace files.

This parameter must be specified even if these resources are not password-protected.

RUNMODE
specifies the openFT start mode:

S or SS	the FJINIT command merely loads openFT.
A or AA	the FJINIT command loads and immediately activates openFT (the FJSTART command is superfluous in this case).
**D	snap dumps can be generated for diagnostic purposes.

"**" stands for "SS" or "AA" with the same meaning as above.

openFT can only be loaded in non-privileged mode for test purposes.

WARNING!
In non-privileged mode, openFT does not check the transfer admission or data access authorization. This means that:

- Transfer requests are accepted and executed even if invalid specifications are made in the TRANSFER-ADMISSION or PROCESSING-ADMISSION.
- The execution of other functions, e.g. follow-up processing or the printing of result lists, however, can be rejected by the system if invalid specifications are detected which openFT has not rejected.

openFT is loaded with the following specifications in non-privileged mode:

N or NS	openFT is loaded in non-privileged mode.
NA	openFT is loaded in non-privileged mode and activated immediately.
NSD	openFT is loaded in non-privileged mode. Diagnostic capabilities are activated.
NAD	As above.

FT-PARMLIB
Name of the openFT parameter library. If no name is entered for this library in FJGEN then openFT uses the default value:
<openft qualifier>.<inst>.PARM
The first two name parts are replaced by OPENFT QUALIFIER and the name of the instance.

If the parameter library does not exist at the time FJGEN is called, openFT creates it with the following content:

– PARM member with the entries:
```
CMD_TRANS=TCP
DSTYPEDEF=PS
LIBTYPEDEF=PO
OPENFT_SVC=211
```
The entry OPENFT_SVC is important if openFT is to perform command encryption using the started openFT subsystem, see section “Providing the OPFT subsystem” on page 92. If the openFT subsystem is not available or not started, the openFT batch job or the started task can only be started if this entry is deleted or invalidated.

– FTADM with the entries
```
"OPENFT USER ID" from FJGEN
Console
```

– FTACADM with the entries
```
"OPENFT USER ID" from FJGEN
Console
```

Details on the parameter library and its members can be found in the section “Setting up the FT parameter library” on section “Setting up the FT parameter library” on page 57).

CMDPORT
Port number of the command client, i.e. the port number of the current openFT instance for connecting the interactive tasks to openFT. CMDPORT is only relevant if the CMD TRANS parameter is not set to VTAM in the PARM member of the openFT parameter library. If no port number is specified here, openFT uses the openFT-specific default port number 1100.

HOST NAME
Host name for the current openFT instance. This information is required for addressing in TCP. The host should be specified directly as an IP address or as a hostname. If multiple openFT instances are to be able to run in parallel with TCP/IP then they must be assigned different IP addresses. Please note that you may only use IP addresses that are defined in your z/OS system’s address space. If you do not specify this value, openFT uses the first IP address that is defined in the z/OS system.

HSM-MCDS NAME
Help file for archiving and restoring (migrating) files. If nothing is specified, openFT sets the default value DFHSM.MCDS.

FJGEN uses the specified installation parameters, for example, to create the following JCL statements for a batch job for loading and starting the openFT load module (these statements are stored in the FJBATCH member of the FT procedure library):

```
//OPENFTF   JOB (A123,B123),                                        (1)
//            CLASS=A,MSGCLASS=A,
//            USER=OPENFT,PASSWORD=OPENFT,                          (2)
//            TIME=1440,REGION=0M
//DLTDMP   EXEC PGM=IEFBR14                                         (7a)
//DELFILE  DD DSN=OPENFTQU.STD.SYSUDUMP.PREV,
//            DISP=(MOD,DELETE,DELETE),
//            SPACE=(CYL,(20,5)),
//            DCB=(DSORG=PS)
//RENAME   EXEC PGM=IDCAMS                                          (9)
//SYSPRINT DD SYSOUT=*
//SYSTSIN  DD *
//SYSIN    DD *
//            ALTER 'OPENFTQU.STD.SYSUDUMP' +
//            NEWNAME ('OPENFTQU.STD.SYSUDUMP.PREV')
//            IF LASTCC = 8 THEN SET MAXCC = 0
//OPENFT   EXEC PGM=OPENFT,TIME=1440,
//            PARM='OPENFTQU,VSN123/SYSDA,A,FTID1,STD,AFFE,1100,'   (3)
//*           openFT V12.1A00 / FJBATCH V121A00
//STEPLIB  DD DSNAME=USERA.OPENFT.LOAD,                             (4)
//            DISP=(SHR,KEEP)
//OPENFTS  DD DSNAME=USERA.OPENFT.NCLOAD,                           (4a)
//            DISP=(SHR,KEEP)
//OPENFT   DD DSNAME=OPENFTQU.STD.CONN,                             (8)
//            DISP=(SHR,KEEP)
//OPFTATT  DD DSNAME=OPENFTQU.STD.OPFTATT,
//            DISP=(SHR,KEEP)
//*DDUADS  DD DSNAME=SYS1.UADS,                                     (5)
//*            DISP=(SHR,KEEP)
//OPFTHSM  DD DSNAME=OPENFTQU.STD.COLLECT.DATA,
//            DISP=(SHR,KEEP)
//MCDS     DD DSNAME=DFHSM.MCDS,DISP=SHR
//SYSIN    DD DUMMY
//SYSOUT   DD DUMMY
//IEBCOUT  DD DUMMY                                                 (6)
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD DSN=OPENFTQU.STD.SYSUDUMP,                            (7)
//         SPACE=(CYL,(20,5)),DISP=(,CATLG),
//         DCB=(DSORG=PS)
```

Explanation:

(1) Jobname = openFT USER ID with appended F "accounting information" as specified for openFT USER ACCOUNT. If multiple openFT instances are to run on your system, then different letters must be appended to the job names.

(2) openFT USER ID and openFT USER PASSWORD

(3) Parameter string consisting of the following start parameters:

Start parameter	Corresponding keywords in PARM
OPENFT QUALIFIER	No corresponding keyword
VOLUME/UNIT	NABVOLUME, NABUNIT, DMP_VOLUME, DMP_UNIT
RUNMODE	RUN_MODE[1]
FT-ID	FJAM_ID [1]
INSTANCE NAME	No corresponding keyword
FT-PASSWORD	FJAM_PASSW[1]
PORT	No corresponding keyword
HOST NAME	No corresponding keyword

[1] These keywords in PARM are only supported for reasons of compatibility.

The start parameters are ignored if you specify the corresponding parameters in FTMSPPAR, see also the section “Setting up the FT parameter library” on page 57

(4) FT-LOADLIB

(4a) FT-NCLOADLIB

(5) This DD statement is created as comment if the SYS1.UADS file exists on your system. If the batch job (or the start procedure for the started task) does not contain a DD statement of this type, openFT cannot check the passwords and account numbers of user IDs via SYS1.UADS (see the section “Linking openFT with data protection products” on page 97). In this case, the following message appears in the job log

```
IEC130I DDUADS DD STATEMENT MISSING
```

If user IDs are to be checked using SYS1.UADS, this DD statement must be activated by removing the comment asterisks.

If user IDs are to be checked using RACF and if a file SYS1.UADS is still present on the system, the comment character must **not** be removed.

If the batch job does contain a DD statement of this type, but the SYS1.UADS file does not exist on your system (i.e. was deleted from the system after the batch job was created using FJGEN), the batch job is terminated with a JCL error and the following message appears in the job log:

```
IEF212I ........ ...... DDUADS - DATA SET NOT FOUND
```

(6) openFT requires this DD statement in order to call the IBM utility IEBCOPY for transferring an entire PO or PDSE data set

(7) DD statement for the generation of a machine-readable dump in the case of system errors; the filename is preceded by the OPENFT QUALIFIER.

(7a) Statements for deleting the dump before the next openFT run.

(8) The instance-specific connection file

(9) With this statement you save the SYSUDUMP file when the batch job is restarted by first having it renamed SYSUDUMP.PREV. As SYSUDUMP does not yet exist when the batch job is started for the very first time, the ALTER command returns condition code 8 when renaming takes place. This is then set to 0.

You can adapt the JCL statements created by FJGEN to the requirements of your own system. Modifications which affect the function of openFT are described in the section “openFT as a job or started task” on page 92.

2.10.2 FJGENPAR - Output installation parameters

Note on usage

User group: FT administrator

This command must be called under TSO.

Functional description

You use the FJGENPAR command to output the openFT installation parameters on screen.

Format

FJGENPAR

Without operands

FJGENPAR outputs the installation parameters of the corresponding openFT instance as defined using FJGEN to the screen.

FJGENPAR only displays the original settings made during the FJGEN run.

Subsequent changes to the settings are not visible here, for instance:

– subsequent editing of the FJBATCH procedure or
– "overwriting" by corresponding parameters in the PARM member of the FT parameter library (such as RUN_MODE), see also page 111.

Example: outputting installation parameters

```
fjgenpar
INSTANCE NAME     : STD
FT-PROCLIB        : OPENFTQU.STD.CLIST
FT-LOADLIB        : USERA.OPENFT.LOAD
FT-NCLOADLIB      : USERA.OPENFT.NCLOAD
VOLUME/UNIT       : VSN123/SYSDA
openFT USER ID    : OPENFT
OPENFT QUALIFIER  : OPENFTQU
FT-ID             : FTID1
FT-PASSWORD       : AFFE
RUNMODE           : A
```

```
FT-PARMLIB          : OPENFTQU.STD.PARM
CMDPORT             : 1100
HOST NAME           :
HSM MCDS            : DFHSM.MCDS
READY
```

For the meaning of the output information, see the example accompanying the description of the FJGEN command (page 104).

2.10.3 FJINIT - Load openFT

Note on usage

User group: FT administrator

This command can be entered in the TSO command mode only.

Functional description

You use the FJINIT command to load and start the openFT load module if openFT is to run as a background process

For information on loading and starting openFT as a started task, see the section “openFT as a job or started task” on page 92.

Tabelle 1:

FJINIT

Without operands

Successful loading of openFT is acknowledged with the following message:

```
JOB useridF (JOBnnnnn) SUBMITTED.
```

The following message is output into the job logging file:

```
FTR4120 OPENFT: INITIATED
```

Notes

– The FJINIT command starts the member FJBATCH of the FT procedure library (see the description of the FJGEN command starting on page 104) defined at installation as a batch job by means of SUBMIT. The job name consists of the OPENFT USERID specified in the FJGEN command plus the letter "F". For technical reasons, the last letter of 8-character user IDs is replaced by an "F". If multiple openFT instances are to run in parallel under a user ID then the job names must end with different last letters. In this case, after running FJGEN, you should replace the "F" in the batch job with another letter (except for L,N, J, Z and P).

– No check is carried out as to whether openFT has already been loaded. If FJINIT is entered twice, the second job is delayed by the job scheduler until the first job has terminated.

– Depending on the openFT start mode, the local openFT instance can also be activated immediately when the openFT load module is loaded and started. In this case, it is not necessary to issue the FTSTART command.

  (See also the RUNMODE parameter in the description of the FJGEN command, page 104.)

2.10.4 FTUPDPAR - Update operating parameters

Note on usage

User group: FT administrator

Functional description

You can use this command to update certain settings in the parameter library while openFT is running. These are the list of FT administrators (FTADM member), the list of FTAC administrators (FTACADM member), the diagnostic settings (DIAGPAR), the code tables in the FNAMECTB member and the installation parameters JOB_MSGCLASS and JOB_JOBCLASS (PARM member). Once you have edited these members you can take over your changes during operation with the FTUPDPAR command.

The changed entry with respect to FNAMECTB is entered in the job log after the FTUPDPAR. If no FNAMECTB member is to be read in, you must remove the current FNAMECTB from the PARM file and call the FTUPDPAR command again. Subsequently no FNAMECTB member which can be accessed exists in openFT.

You are recommended to stop openFT before calling FTUPDPAR with FTSTOP and then to restart it with FTSTART.

Format

FTUPDPAR

Without operands

3 Operation of openFT

This chapter contains information on the subject of administration, security and control and monitoring functions.

3.1 Optimizing the operating parameters

The proposals listed below suggest a number of ways in which the FT administrator can optimize FT operation by modifying the operating parameters. It is always advisable to alter only one operating parameter at a time, so that the precise effects of the change can be observed.

3.1.1 Interdependencies for optimized parameterization

The optimum settings for operating parameters depend on several different constraints:

– load levels of the local and remote systems,
– load level in the network,
– line transfer rates in the network,
– network structure (connection paths reserved for FT or shared paths for FT and dialog),
– incorporation of gateway computers,
– type, performance or generation of the transport system used,
– average size of files to be transferred,
– number of files to be transferred (e.g. per day).

In some instances, these boundary conditions are themselves subject to dynamic change (load levels for example), so it is not possible to calculate in advance the optimized values for a particular installation.

3.1.2 Achieving optimized operation

Experience has shown that the most suitable parameter settings can only be achieved in stages.

Initially the openFT default values should be left unchanged. In most cases it will be possible to run file transfers satisfactorily using these parameter values.

If not, however, as a second step an improvement can be sought by changing **one** of the parameter values. It is normally not advisable to change more than one parameter at a time as otherwise there is no way of ascertaining the precise effect of each change.

If satisfactory operation of the FT system has still not been achieved, the FT administrator can repeat the second step, changing a different parameter.

The FT administrator can control the operation of the FT system using the parameters PROCESS-LIMIT, CONNECTION-LIMIT, TRANSPORT-UNIT-SIZE and MAX-REQUEST-LIFETIME, see the following table:

Problem	**Suggested solution**
Poor dialog response times	1. Reduce TRANSPORT-UNIT-SIZE 2. Reduce CONNECTION-LIMIT
Computer overloaded, network load not yet optimized	1. Set PROCESS-LIMIT to 1 or 2 2. Increase TRANSPORT-UNIT-SIZE 3. Reduce CONNECTION-LIMIT
Computer and network overloaded	1. Set PROCESS-LIMIT to 1 or 2 2. Reduce CONNECTION-LIMIT
Throughput inadequate	1. Increase TRANSPORT-UNIT-SIZE 2. Under TCP/IP: Set RFC1006 transport protocol (see FTADDPTN command)
Prolonged requests block other requests	1. Increase CONNECTION-LIMIT
Requests to a particular partner system use up all resources	1. Set the partner system to low priority with PRIORITY=*LOW 2. Increase CONNECTION-LIMIT 3. Set REQUEST-PROCESSING=*SERIAL for the corresponding partner system.
Requests from partner systems (inbound requests) use up all resources	1. Increase CONNECTION-LIMIT
Requests are present in the request file for a very long period without being processed.	1. Set MAX-REQUEST-LIFETIME

The command used for this purpose is FTMODOPT. These parameters are discussed in the sections below. In addition, the effect of changing the parameters is also described.

3.1.3 Changing the PROCESS-LIMIT operating parameter

The PROCESS-LIMIT parameter defines the maximum number of tasks that may be used for processing file transfer requests. The number of file transfer requests per task handled simultaneously can be expressed as follows:

$$\frac{\text{CONNECTION-LIMIT}}{\text{PROCESS-LIMIT}}$$

CONNECTION-LIMIT is the maximum number of parallel transport connections that can be used to execute requests.

If the PROCESS-LIMIT value remains fixed and the value of CONNECTION-LIMIT is increased, then proportionately more transport connections are available for each task and therefore more requests can be processed per task. The reduction of the PROCESS-LIMIT value where CONNECTION-LIMIT remains constant achieves the same effect. If the value of the quotient is reduced (by reducing CONNECTION-LIMIT or increasing PROCESS-LIMIT), a smaller proportion of transport links is available per task. Consequently, fewer requests can be processed per task.

If the number of requests awaiting processing exceeds the value of the quotient but the number of tasks assigned has not reached the PROCESS-LIMIT value, then another task is initiated.

Higher PROCESS-LIMIT:

– fewer wait times for input/output

– better use of potentially underutilized computer resources

Lower PROCESS-LIMIT:

– reduced load on the local system

3.1.4 Changing the CONNECTION-LIMIT operating parameter

The CONNECTION-LIMIT parameter defines the maximum number of transport connections to be used in the execution of file transfer requests. Since the processing of a request always requires a new transport connection to be set up, CONNECTION-LIMIT also defines the maximum number of requests the system can process in parallel.

A third of the connections is reserved for outbound requests and a third for inbound requests. The remaining third can be used for inbound or outbound requests as required. You may therefore have to increase the value of CONNECTION-LIMIT to achieve the required throughput to your openFT partners.

Higher CONNECTION-LIMIT:

– increased data throughput

– better use of potentially underutilized processor capacity.

Lower CONNECTION-LIMIT:

– reduced load on the local system and network, and hence less or even no impact upon interactive operation.

3.1.5 Changing the TRANSPORT-UNIT-SIZE operating parameter

The TRANSPORT-UNIT-SIZE parameter defines the maximum length of the message transmitted to the transport system by openFT. Message flow control ensures that only a specific number of messages are being transmitted across the network at any one time. The TRANSPORT-UNIT-SIZE parameter enables the administrator to control the amount of FT data per connection present in the network at a particular time. The value specified for TRANSPORT-UNIT-SIZE can be changed by the remote system or by the transport system (maximum message length).
A maximum value of 65535 is recommended for TRANSPORT-UNIT-SIZE. This value is the default value after installation.

Higher TRANSPORT-UNIT-SIZE:

– increased data throughput

– reduced load on the local system since fewer calls to the transport system are necessary.

Lower TRANSPORT-UNIT-SIZE:

– reduced load on the network

– the time required to transmit an FT message across a communication link is reduced, which in turn decreases the wait time for messages from other users. For slow communication links, response times can, for example, be improved in interactive mode.

3.1.6 Setting the MAX-REQUEST-LIFETIME operating parameter

The MAX-REQUEST-LIFETIME parameter is used to set a global limitation for the lifetime of openFT requests. The maximum lifetime applies to both inbound and outbound requests and is specified in days.

When this period expires, openFT deletes the request by executing the NCANCEL command internally.

3.2 Administering code tables

Available character sets and code tables supplied

The following character sets are completely integrated in openFT:

Name of the CCS	Meaning
ISO88591 to ISO8859B and ISO8859D to ISO8859G	for the ASCII tables ISO8859-1 to ISO8859-11 and ISO8859-13 to ISO8859-16
ISO646	for the international 7-Bit ASCII table
ISO646DE	for the german 7-Bit ASCII reference version
EDF041 to EDF04A and EDF04D to EDF04F	for the EBCDIC tables DF04-1 to DF04-10 and DF04-13 and DF04-15
EDF03IRV	for the international 7-Bit EBCDIC table
EDF03DRV	for the german 7-Bit EBCDIC table
UTF16	for Unicode with UTF-16 coding (platform-specific endian)
UTF8	for Unicode with UTF-8 coding
UTFE	for Unicode with the UTF-E coding
UTF16LE	for Unicode with UTF-16 coding (little-endian)
UTF16BE	for Unicode with UTF-16 coding (big-endian)
UTFEIBM	for Unicode with the UTF-EBCDIC coding defined by IBM
CP1252	for ANSI character set with Euro symbol defined by Microsoft (s.o.)
IBM1047	for the OpenExtensions EBCDIC character set defined by IBM
CP850	for the OEM character set defined by Microsoft

In addition, the code tables IBM037, IBM273 and IBM500 are stored in <openFT installation directory>.OPENFT.SYSCCS on installation. These tables were previously named FTCP037, FTCP273 and FTCP500.

Creating code tables for custom character sets

When migrating from openFT Version V9 (or earlier) to V12, custom code tables must be converted to the new format. It should be noted that the code tables in V9 converted in both directions between EBCDIC.DF.04-1 and the relevant custom code. In V12, the custom code must be mapped to UTF-16, which corresponds to mapping to ISO8859-1 (each character being prefixed by a 00 byte). Newly created code tables now allow characters to be represented that are not contained in ISO8859-1.

You must save the code tables as members in <openft qualifier>.<inst>.SYSCCS. This PO library is empty after installation.

The member name is the CCS name of the associated character set. Assembly is no longer required.

Structure of a code-conversion table

The text file must have the following structure:

- The first line starts with a '#'.

  The second character is an blank. The remainder of the line contains a comment which characterizes the code contained.

- The second line contains an alphabetic character which can at present only have the value 'S'. 'S' stands for single-byte code, i.e. a character is always 1 byte in length.

- The third line contains three numbers.

  The first number is a 4-digit hexadecimal number. This defines the substitution character to be used if a Unicode character cannot be mapped to the code.

  The second number is currently always '0'.

  The third number is a decimal number which defines the number of code pages that follow. It currently always has the value '1'.

- The following lines define the code pages and have the following structure:

  – The first of these lines contains the number of the code page in the form of a two-digit hexadecimal number. Currently, only code page 00 is permitted.

  – Each of the subsequent lines contains the assignment of a character to the corresponding 8-bit code position. A character is represented by its UTF-16 code in the form of a four-digit hexadecimal number. The values are arranged in 16 lines, each of which contains 16 4-digit hexadecimal numbers with no spaces.

Example for ISO8859-15 (Western Europe with Euro symbol)

```
# Encoding file: iso8859-15, single-byte
S
003F 0 1
00
000000010002000300040005000600070008000900A000B000C000D000E000F
001000110012001300140015001600170018001900A001B001C001D001E001F
002000210022002300240025002600270028002900A002B002C002D002E002F
003000310032003300340035003600370038003900A003B003C003D003E003F
004000410042004300440045004600470048004900A004B004C004D004E004F
005000510052005300540055005600570058005900A005B005C005D005E005F
006000610062006300640065006600670068006900A006B006C006D006E006F
007000710072007300740075007600770078007900A007B007C007D007E007F
008000810082008300840085008600870088008900A008B008C008D008E008F
009000910092009300940095009600970098009900A009B009C009D009E009F
00A000A100A200A300AC00A5016000A7016100A900AA00AB00AC00AD00AE00AF
00B000B100B200B3017D00B500B600B7017E00B900BA00BB0152015301780BF
00C000C100C200C300C400C500C600C700C800C900CA00CB00CC00CD00CE00CF
00D000D100D200D300D400D500D600D700D800D900DA00DB00DC00DD00DE00DF
00E000E100E200E300E400E500E600E700E800E900EA00EB00EC00ED00EE00EF
00F000F100F200F300F400F500F600F700F800F900FA00FB00FC00FD00FE00FF
```

3.3 Administering requests

FTMODREQ	Modify the order and priority of outbound requests within the request queue
NCANCEL	– Delete FT request from the request queue – Abort file transfer while in progress
NSTATUS	Information on FT requests and their properties
FTMODPTN	Activate or deactivate locally submitted requests for a particular remote system (STATE operand)

3.4 Administering partners

Partner systems can only be administered if they are entered in the partner list. You have two options to do this:

- You enter the partner with name and address (named partner).
- You enter the partner only with address but without name (registered dynamic partner). In this case, you have to note some details, see section „Registered dynamic partners“.

I.e. free dynamic partners (partners which are not entered in the partner list) cannot be administered. Please refer to the openFT manual "Concepts and Functions" for more information on the partner concept.

openFT offers the FT administrator the following commands for the administration of partner systems:

FTADDPTN	Add new partner system entries to the partner list
FTMODPTN	Modify partner system entries in the partner list
FTREMPTN	Remove partner systems from the partner list
FTSHWPTN	View information on partner systems in the partner list and back up the partner list (page 130)
FTMODOPT	Define the global FTAC security level, enable/disable dynamic partners

Registered dynamic partners

All partners that are entered only with their addresses but without names in the partner list are registered dynamic partners. You enter partners of this type in the partner list as follows:

```
FTADDPTN PARTNER-NAME=*NONE
                ,PARTNER-ADDRESS=address,<other attributes>.
```

I.e., you assign one or more attributes with a value other than the default, e.g. TRACE=*ON.

Please note:
– Security level based on the partner setting (SECURITY-LEVEL=*BY-PARTNER-ATTRIBUTES) is the default setting for free dynamic partners and therefore does not count as a differently set attribute.
– In contrast, security level based on the operating parameter setting (SECURITY-LEVEL=*STD; default setting for the FTADDPTN command) is a differently set attribute.

If you reset all the attributes for a partner of this type to the default values with FTMODPTN then this partner is removed from the partner list and becomes a free dynamic partner.

3.4.1 Backing up the partner list

You can back up the entries in the partner list by means of the FTSHWPTN command:

- FTSHWPTN outputs the partner entries in the form of FTMODPTN commands. To do this, specify the OUTPUT=*SYSLST(LAYOUT=*ZOS-PROC) operand.

3.5 Security in FT operation

The following functions offer an even higher level of security in file transfer:

- Authentication
- Extended authentication check
- Encryption for file transfer
- Protection mechanisms against data manipulation

Note

openFT is able to exchange encrypted outbound file contents with a Secure FTP server if openFT-CR is installed on the openFT side and the FTP server supports the TLS (**T**ransport **L**ayer **S**ecurity) protocol. The Secure FTP server makes its key and the certificate available to the openFT instance for encryption purposes. No mutual authentication is carried out.

3.5.1 Authentication

3.5.1.1 Instance identifications

The instance ID must be unique throughout the network irrespective of case.

Local instance identification

During installation, the name of the computer in the local network is defined by default as the instance ID. If it cannot be guaranteed that this name is unique in the network then you must change the instance ID. To do this, use the following command:

FTMODOPT, IDENTIFICATION=instance-id

Instance identification of partners

Store instance IDs of partner systems in the partner list using the IDENTIFICATION parameter of the FTADDPTN command, or FTMODPTN. With the aid of the partner systems' instance IDs, openFT manages the resources assigned to those partners, such as request queues and cryptographic keys.

3.5.1.2 Creating and managing local RSA key pairs

RSA keys are used for authentication as well as for the negotiation of the AES key with which the request description data and file contents are encrypted.

You can use the following commands to generate and manage local RSA key.

FTCREKEY	creates an RSA key pair set for the local openFT instance
FTSHWKEY	shows the attributes of all keys in the local system
FTUPDKEY	updates the public keys
FTDELKEY	deletes local RSA key pair sets
FTMODKEY	modifies RSA key attributes
FTIMPKEY	imports RSA keys

Key pair attributes

Each RSA key pair consists of a private and a public ksey. There can exist up to three key pair sets each consisting of three key pairs with lengths of 768, 1024, 2048. The FTCREKEY command generates new key pairs for each of these lengths.

Private keys are internally administered by openFT. Public keys are stored on the configuration user ID of the openFT instance (standard: $SYSFJAM), under the following name:

```
<inst>.SYSPKF.R<key reference>.L<key length>
```

The key reference is a numeric designator for the version of the key pair.

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 and z/OS, ISO8859-1 for Unix systems and CP1252 for Windows systems.

A key of length 2048 is used by default for encryption. You can modify this setting using the FTMODOPT command.

Storing comments

In a SYSPKF.COMMENT file on the configuration user ID of the openFT instance, you can store comments, which are written in the first lines of the public key files when a key pair set is created. Comments could, for example, contain the contact data for the FT administrator on duty, the computer name, or similar information that is important for partners. The lines in the SYSPKF.COMMENT file may be a maximum of 78 characters in length. Using the FTUPDKEY command, you can import updated comments from this file into existing public key files at a later time.

Updating and replacing keys

If a public key file has been unintentionally deleted or otherwise manipulated, you can re-create the public key files of the existing key pair sets using FTUPDKEY.

If you want to replace a key pair set with a completely new one, you can create a new key pair set using FTCREKEY. You can identify the most current public keys by the highest value key reference in the file name. openFT supports a maximum of three key pair sets at a time. The existence of several keys should only be temporary, until you have made the most current public keys available to all the partner systems. Afterwards, you can delete the key pair sets no longer needed using FTDELKEY.

If the openFT administrator is not the same as the system administrator, it must be ensured that this administrator has access to the SYSPKF files and the <inst>.SYSKEY library on the OPENFT QUALIFIER of the openFT instance. This can be done, either by assigning operating system-specific access rights or by setting up corresponding FTAC admission profiles.

3.5.1.3 Importing keys

You can use the FTIMPKEY command to import the following keys:

- Private keys that were generated with an external tool (i.e. not via openFT). When importing a private key, openFT generates the associated public key and stores it under the configuration user ID of the openFT instance, see “Key pair attributes” on page 132. This key can be used in the same way as a key generated with FTCREKEY and distributed to partner systems.
- Public keys of partner instances. These keys must have the openFT key format (syspkf), i.e. they must have been generated by the partner's openFT instance. openFT stores the key in the SYSKEY library, see “Managing the keys of partner systems” on page 134.

Every imported key pair contains a unique reference number. RSA keys with the supported key lengths are imported (768, 1024 and 2048 bits).

openFT supports key files in the following formats:

- PEM format (native PEM)

  The PEM-coded files must be present in EBCDIC format.

- PKCS#8 format encrypted without password phrase or after v1/v2 with password phrase (PEM-coded).

  You must specify the password phrase used for encryption in the password parameter when you perform the import.

- PKCS#12 v1 format in the form of a binary file. The file is searched for a private key and any non-supported elements (e.g. certificates, CRLs) are ignored during the import. 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. The first private key that is found in the file is imported. Any others are ignored.

  You must specify the password phrase used for encryption in the password parameter when you perform the import.

3.5.1.4 Managing the keys of partner systems

The public keys of the partner systems are to be stored in z/OS as members in the **<inst>.SYSKEY** library under the OPENFT QUALIFIER of the local openFT instance.

The partner name of the partner system as defined in the partner list must be selected as the element name.

You can import the public key of a partner system in the following ways:

- You can specify the name of the key file in the FTIMPKEY command. When you perform the import, openFT checks whether there is a partner list entry with the instance ID that is stored in the key file. If there is then openFT stores the key under the partner's name in the SYSKEY library.
- You can use the tools available in the operating system to copy the key file in the correct format to the SYSKEY library and save it there under the partner's name.

If an updated public key is made available by the partner instance, the old key must be overwritten by it.

You can use the command FTSHWKEY ...SELECT=*PAR(PARTNER-NAME=...) to display the keys of partner systems and filter on expiration date.

While the SYSKEY library is open for updating, openFT is unable to perform any authentication of inbound requests and new requests are rejected. You should therefore make sure that the library is not open for long, for example by entering the updated members in SYSKEY via openFT. If you stop openFT to work on SYSKEY (with FTSTOP) then new restartable inbound requests are stored in the partner systems and are subsequently processed automatically.

Modifying the keys of partner systems

You can use the FTMODKEY command to modify the keys of partner systems by specifying an expiration date or modifying the authentication level (1 or 2):

- If you specify an expiration date then it is no longer possible to use the key once this date has expired.

- If you set authentication level 2 then openFT also performs internal checks. Level 2 is supported for all openFT partners as of Version 11.0B. Level 1 authentication attempts to this partner are rejected.

You can make these settings for a specific partner or for all partners, as you require, and modify them subsequently if necessary.

3.5.1.5 Distributing the keys to partner systems

Distributing the public key files to your partner systems should take place by secure means, for example by

– distribution by cryptographically secure e-mail
– distribution on a CD (by courier or by registered mail)
– distribution via a central openFT file server, the public keys of which are in the partners' possession

If you transmit your public key files to partner systems using Unix or Windows operating system, you must ensure that these files are re-coded from EBCDIC.DF04-1 to ISO 8859-1 or CP1252 (e.g. by transferring them as a text file via openFT).

The public key file of your local openFT instance is stored in the partner system in the following location:

- For partners with openFT (BS2000), as a type D PLAM element in the **SYSKEY** library, the configuration user ID of the partner instance. The partner name allocated for your openFT instance in the remote network description file or in the remote partner list must be selected as the element name.
- For partners with openFT (Unix systems), in the **/var/openFT/<instance>/syskey** directory. The instance ID of your local openFT instance must be selected as the file name. The file name must not contain any uppercase letters. If the instance ID contains uppercase letters, these must be converted to lowercase in the file name.
- For partners with openFT (Windows), in the directory <**openFT installation directoy>\var\<Instance>\syskey**, in newer Windows versions such as Windows 10 in **%ProgramData%\Fujitsu Technology Solutions\openFT\var\std\syskey**.
  The instance ID of your local openFT instance must be selected as the file name.
- For partners with openFT (z/OS), as a PO element in the **<inst>.SYSKEY** library. The partner name allocated for your openFT instance in the remote network description file or partner list must be selected as the element name.

3.5.2 Extended authentication check

openFT partners using openFT from version 8.1 onwards, support the authentication mechanism (see page 131). If the local system has a public key of the partner at its disposal, the partner's identity is checked by cryptographic means.

For partner systems that do not work with authentication, inbound requests are checked with the aid of the instance identification, in order to ascertain whether the calling system has a valid entry in the partner list. openFT offers via extended sender checking the possibility of checking not only the instance identification, but also the transport address.

The extended sender checking can be globally enabled for openFT partners or just for specific partners:

- globally, using
  FTMODOPT ... PARTNER-CHECK=*TRANSPORT-ADDRESS
- only for specific partners, using
  FTADDPTN ... PARTNER-CHECK=*TRANSPORT-ADDRESS or
  FTMODPTN ... PARTNER-CHECK=*TRANSPORT-ADDRESS

The global setting is valid for all partners with the value PARTNER-CHECK=*BY-FT-OPTIONS (default in the FTADDPTN).

In the case of FTAM and FTP partners, the sender check operates exclusively via the transport address. Consequently the "extended sender verification" attribute is ineffective for FTAM and FTP partners and is also not displayed.

Extended sender verification is of no relevance for dynamic partners because these are always identified via the transport address.

If the authentication check returns a negative result, the request is rejected.

3.5.3 Encryption for file transfer

openFT supports for openFT partners the encryption of the data sent and received in the process of setting up the connection and processing a file transfer request. The partners involved in file transfer automatically negotiate encryption and use of the appropriate public key in the process of connection set-up.

If possible, openFT uses the RSA/AES procedure with a AES key length of 256 bits for encryption. In the case of connections with older partners, RSA/AES with 128-bit AES key length or RSA/DES may also be used. In all cases, the most secure of the procedures that are supported by both partners is used. In order to increase the security you can define additionally a RSA minimum key length and/or a AES minimum key length (FTMODOPT command, KEY-LENGTH= operand).

openFT automatically encrypts the request description data if both partners support this functionality, there is an RSA key pair set in the local system and encryption has not been explicitly disabled (command FTMODOPT ...,KEY-LENGTH=(RSA-MINIMUM=0)). You can use the SHOW-FT-OPTIONS command to check the key length that is currently being used (output parameter KEY-LEN). You can set the key length required for the RSA key via the operating parameters (FTMODOPT command KEY-LENGTH parameter).

Using the FTCREKEY command, the FT administrator must create at least one key pair set, upon which the encryption will be based and carried out. Alternatively, the administrator can also import a key pair of the configured key length using
FTIMPKEY.

If, in addition to the request description data, the file content is to be encrypted for transfer by openFT, then the optional openFT-CR component must be installed on both FT systems involved.

If one of the two systems is not capable of handling encrypted file transfers, the request is rejected with the message FTR2051 (User data encryption not possible for this request) or with FTR2113 (encryption is not possible in remote system).

For legal reasons, openFT-CR is not available in all countries.

Forcing encryption

Encryption of the file contents is optional and is usually requested during the transfer request. However, you can also use the operating system parameters to force encryption (mandatory encryption). To do this, use the ENCRYPTION-MANDATORY operand in the FTMODOPT command.

Mandatory encryption can be set differently for different operations (only inbound, only outbound or all requests). The settings apply to file transfer requests via the openFT protocol as well as for administration requests. Inbound FTP requests are rejected because no encryption is permitted. File management continues to be performed irrespective of the settings. In addition, the following applies:

- If outbound encryption is activated then the file content is encrypted on outbound requests even if no encryption is demanded in the request itself. If the partner does not support encryption (e.g. because it is deactivated or because openFT-CR is not installed) then the request is rejected.
- If an unencrypted inbound request is to be processed while inbound encryption is activated, then this request is rejected.

3.5.4 Protection mechanisms against data manipulation

During communications with openFT partners, openFT as of V8.1 implicitly checks the integrity of the transferred data. For requests with unencrypted file content, the integrity of the request description data is checked. For requests with encryption, the integrity of the transferred file content is also checked. If an error is detected, restartable requests attempt a new transfer. Non-restartable requests are aborted.

In this way it is possible to detect and prevent malicious manipulations of the transferred data (e.g. in insecure public networks such as the Internet).

Errors on the physical transfer channels are identified and rectified by the communication system itself. No data integrity check at openFT level is required for this.

3.6 Monitoring and controlling FT operation

Fetch information on the FT system

The FT administrator uses the following commands to obtain information on the system:

FTSHWOPT	Information on operating parameters
FTSHWPTN	Information on partner systems
FTSHWLOG	Information on log entries
NSTATUS	Information on file transfer status
FTSHWINS	Information on openFT instances
FTSHWMON	Show monitoring data from openFT operation

The FTSHWOPT command furnishes information on the current settings of the operating parameters.

FTSHWPTN yields information on the partner systems and their associated properties, e.g., names, addresses, security levels for FTAC, and so on.

To support automatic monitoring, some events which are not direct responses to user input are reported by openFT via console messages. More detailed information on this topic can be found in the section “Console messages for automatic monitoring” on page 143.

The command FTSHWLOG can be used to display the logs of file transfer requests.

NSTATUS enables the FT administrator to retrieve information on all file transfer requests in his or her system, even when the FT system is stopped.

Using FTSHWINS, the FT administrator can find out which openFT instances exist in the system and have their characteristics and status displayed. FTSHWINS only works if openFT has been started as a subsystem.

FTSHWMON outputs the monitoring values from openFT operation. To do this, monitoring must be activated by means of FTMODOPT.

3.6.1 FT logging

The following 3 commands are available for the FT logging function:

FTDELLOG	– Deleting log records
	– Deleting offline log files
FTMODOPT	– Switching on/off the logging function and define the scope of logging
	– Changing the log file
	– Define whether log entries are to be regularly deleted and, if necessary, specify the deletion interval.
FTSHWLOG	– View information on log entries
	– Listing log file names

openFT can record the results of all file transfer requests, irrespective of whether the initiative is in the local or the remote system (outbound and inbound requests, respectively). The information on each successfully completed or aborted request is recorded in an FT logging record. The file consisting of these logging records thus represents a complete, uninterrupted documentary record of FT operation over a prolonged period of time. openFT writes the logging records into the log file <inst>.SYSLOG-SYSLOG.Lyymmdd.Lhhmmss under the OPENFT QUALIFIER of the openFT instance.

yy = year, 2-digit.
mm = month, 2-digit.
dd = day, 2-digit.
hh = hour, 2-digit.
mm = minute, 2-digit.
ss = second, 2-digit.

The date and time designate the time (GMT) at which the log file was created. This suffix makes it possible to distinguish between the current and offline log files.

If the openFT qualifier (OPENFT QUALIFIER in FJGEN) is more than 11 characters in length then the suffix is truncated. If a "Second Level Qualifier" is defined for logging in the z/OS parameter library (LOGFILE_2ND_Q, see page 64) then the suffix is truncated if, together, the openFT Qualifier and this Second Level Qualifier are longer than 23 characters. If the sum of these lengths is greater than 31 characters then the entire suffix is omitted. In this case, it is no longer possible to change the log file.

Changing the log file and administering offline log files

You can change the log file using the command FTMODOPT LOGGING=*CHANGE-FILES. This closes the current log file which is nevertheless retained as an offline log file. For the following log records, a new log file is created with the current date in the suffix. You can change the log file several times and therefore manage multiple offline log files.

This change-over has the following benefits:

– Faster access to logging information due to smaller log files.

– Improved administration of log records through regular change-overs and back-ups of the offline log files.

– Possibility of performing extensive searches in the offline logging information without affecting ongoing openFT operation.

Saving and deleting log records

As one of your duties as FT administrator, you should regularly create backups of the log records from the current log file or from the offline log file(s) as a file in CSV format or on tape, for example and then delete the log records or offline log file(s) with the FTDELLOG command.

In this way you have a complete, uninterrupted log at your disposal for documentation purposes, while at the same time no storage capacity is wasted. Bear in mind the assigned file size of the current log file does not change when you delete log records, but the space formerly occupied by the records you delete is released within the file.

Viewing the contents of a log record

The information content of the FT logging records includes:

– date and time of request processing,
– an acknowledgment indicating correct completion of a request, or the reason for request rejection or abort,
– the direction of file transfer,
– the name of the partner system involved in file transfer,
– TSN and user ID of the request initiator for requests submitted in the local system; only *REMOTE is entered for remote request initiators,
– the user ID under which the request was handled or should have been handled,
– the name of the file,
– the global request ID for inbound requests,
– if an abort occurs, additional information on the cause.

The FT administrator can use the FTSHWLOG command to output all FT logging records of his/her system. Two formats are available for the output: a format that is suitable for listings, and a CSV format that is optimized for further processing. He/she can also choose between a brief overview or a long detailed output and use NUMBER=*POLLING(..) to repeat the output of the new log records at regular intervals.

If the FTAC functionality is being used, the logging records relevant for FTAC are saved in the same file.

Modifying logging settings

You can set the scope of the logging functions and define the times and intervals for the automatic deletion of log records.

Setting the scope of logging

You set the scope of logging with the LOGGING=SELECT(...) operand in the FTMODOPT command.

You can set the scope of FT, FTAC and administration function logging differently. Following installation, full logging is set.

Setting the automatic deletion of log records

You can set the intervals for the automatic deletion of log records in the FTMODOPT command by setting the operand DELETE-LOGGING=*PAR(..). This setting deletes log records as of a defined minimum age at regular intervals and at a specified time. This automatic delete function is only active if openFT is started. If openFT is not started at a scheduled delete time then the delete operation is not performed on the next start-up.

Following installation, the automatic deletion of log records is disabled. You should only enable this function if you do not require the uninterrupted recording of log records.

3.6.2 The openFT job log

Beside the log file the openFT job log also contains information which may be useful for the FT administrator. Some messages are output **only** to the openFT job log; often, however, the chronological order of the messages contained in the job log is useful in the diagnosis of errors during FT operation. The information contained in the openFT job log is described in the appendix on page 191.

3.6.3 Console messages for automatic monitoring

Messages are usually issued as responses to administration commands. There are, however, also some messages which are not (or not exclusively) issued by administration commands. When errors occur on accessing the request queue or the partner list, openFT generates normal DMS error messages.

To support automatic monitoring, some events which are not direct responses to user input are reported by openFT via a console message. Depending on which events are involved, further actions can then be initiated by automatic operators such as Omnis-Prop, HLL-Prop, etc. Console messages can also be used to generate SNMP traps for automatic FT monitoring using SNMP.

The console messages for automatic monitoring occupy the message code range from FTR0300 to FTR0399. They can be activated and deactivated with FTMODOPT CONSOLE-TRAPS=*ON/*OFF. openFT outputs these messages asynchronously. This means that the output is also dependent on the settings for asychronous messages in the PARM library (see “Structure of the PARM member” on page 60).

Messages for monitoring partner systems

FTR0301 OPENFT: Partner '(&00)' entered state NOCON

FTR0302 OPENFT: Partner '(&00)' entered state ACTIVE

FTR0303 OPENFT: Partner '(&00)' entered state LUNK

FTR0304 OPENFT: Partner '(&00)' entered state RUNK

FTR0305 OPENFT: Partner '(&00)' entered state INACT

FTR0306 OPENFT: Partner '(&00)' entered state AINACT

FTR0307 OPENFT: Partner '(&00)' may be unreachable

FTR0308 OPENFT: Partner '(&00)' does not allow more inbound requests

FTR0309 OPENFT: Partner '(&00)' added

FTR0310 OPENFT: Partner '(&00)' removed

FTR0311 OPENFT: Partner '(&00)') entered state LAUTH

FTR0312 OPENFT: Partner '(&00)' entered state RAUTH

FTR0313 OPENFT: Partner '(&00)' entered state DIERR

FTR0314 OPENFT: Partner '(&00)' entered state NOKEY

FTR0315 OPENFT: Partner '(&00)' entered state IDREJ

Messages for monitoring openFT

FTR0320 OPENFT: abnormal termination initiated

FTR0360 OPENFT: openFT control process started

FTR0361 OPENFT: openFT control process terminated

Messages for monitoring the request queue

FTR0330 OPENFT: Request queue 85 percent full

FTR0331 OPENFT: At least 20 percent of request queue unoccupied

Messages for monitoring requests

FTR0340 OPENFT: Transfer '(&00)' successfully completed

FTR0341 OPENFT: Transfer '(&00)' terminated with error

3.6.4 Monitoring with openFT

openFT provides the option of monitoring and recording a range of characteristic data for openFT operation. The data falls into three categories:

- Throughput, e.g. total network throughput caused by openFT
- Duration, e.g. processing time for asynchronous jobs
- State, e.g. number of requests currently queued

You must be an FT administrator in order to activate, deactivate or configure monitoring.

As soon as monitoring is activated, any user can call up the data and output it based on certain criteria.

3.6.4.1 Configuring monitoring

You configure monitoring using the FTMODOPT command and the MONITORING= operand. The following options are available:

- Activating and deactivating monitoring
- Selective monitoring based on the partner type
- Selective monitoring based on the request type

Once you have chosen your settings, they are retained until you change them explicitly. This means that they are also not changed if you reboot the computer.

You can check the current settings with FTSHWOPT. The MONITOR row indicates whether monitoring is activated and shows any criteria used for selection.

3.6.4.2 Showing monitoring data

If monitoring is activated the monitoring data can be called up on the local system or from a remote system.

Outputting monitoring data on the local system

Use the command FTSHWMON to show monitoring data locally.

FTSHWMON outputs the monitoring data in the form of tables that you can further process as required either programmatically or using an editor.

When you call FTSHWMON you can select specific monitoring data for output, whether or not output is formatted and the time interval at which output is performed. You can also specify the output medium. You can find details on the values output in the description of the FTSHWMON command.

Showing monitoring data on remote Unix or Windows systems

The monitoring data can also be shown in the openFT Monitor on a remote Unix or Windows system. To do this, you set up a special admission profile that is specified when the openFT monitor is called and causes only the monitoring values to be read and transferred. The admission profile uses the keyword *FTMONITOR as a preprocessing command and is set up as follows:

```
FTCREPRF NAME=MONITOR,TRANSFER-ADMISSION=ONLYFTMONITOR -
       ,FILE-NAME=*EXPANSION('|*FTMONITOR ') -
       ,FT-FUNCTION=(*TRANSFER-FILE,*FILE-PROCESSING)
```

ONLYFTMONITOR is the (freely selectable) FTAC transfer admission that must be specified when the openFT Monitor is called. Alternatively, this transfer admission can also be specified in an ft or ncopy command used to transfer monitoring data in a Unix or Windows system.

You will find details in the manual "openFT (Unix and Windows systems) - Installation and Operation".

3.7 Administrating and controlling FTAC functions

As the FTAC administrator, you are responsible for the following tasks:

- Administrating admission sets
- Administrating admission profiles
- Saving and migrating the FTAC environment
- Administering the FTAC logging function

WARNING!

Note that openFT-AC is only effective for connected products such as openFT. If other file transfer products without an openFT-AC connection are also being used, a more comprehensive and coordinated security concept would be advisable.

The FTAC administrators of an openFT instance are defined by means of an entry in the FTACADM member of the FT parameter library PARM.The FTAC file SYSFSA is automatically created in order to store FTAC administration data, such as admission sets, admission profiles, etc. (see section “Internal openFT data sets” on page 193).

3.7.1 Administrating admission sets

For the administration of admission sets, openFT-AC offers the FTAC administrator the following commands:

FTMODADS	Modify admission sets
FTSHWADS	Show admission sets

As the FTAC administrator, you are responsible for the following tasks:

- You define the standard admission set using the command FTMODADS USER-IDENTIFICATION = *STD.

  Following the installation of FTAC, all values of the standard admission set are to 0, i.e. no file transfer is possible! As FTAC administrator, you should therefore adapt the standard admission set to the protection requirements on your processor.

  The user can override the entries in the standard admission set only if you, as FTAC administrator, modify the admission set of the user accordingly or if you set up a privileged FT profile.

- You can display admission sets of all users of the system using the FTSHWADS command.

  The entries made by the FTAC administrator are listed under MAX-ADM-LEVELS, the user entries under MAX-USER-LEVELS. The smaller value is valid in each case.

- For each user in the system, you can assign an individual admission set or modify an existing one using FTMODADS.

3.7.2 Administrating admission profiles

For the administration of admission profiles, openFT-AC offers the FTAC administrator the following commands:

FTCREPRF	create admission profile
FTDELPRF	delete admission profile
FTMODPRF	modify admission profile
FTSHWPRF	show admission profile

The FTAC administrator has the option of modifying foreign admission profiles:

– The administrator can create admission profiles for foreign users with the FTCREPRF command. However, certain restrictions apply (see page 148).

– He can view them with the command FTSHWPRF. The transfer admission of an admission profile is not output. This means that the FTAC administrator does not have access rights to the files of foreign user IDs.

– He can delete them with the command FTDELPRF. This is the most radical of all options which should only be used in extreme cases and with good reason and upon consultation with the owner of the profile.

– He can privilege them with the command FTMODPRF, or conversely revoke privileges.

– He can also modify them with FTMODPRF. If the FTAC neither possesses the SU privilege nor specifies the complete USER-ADMISSION including the account and the password of the owner of the profile then the access to the admission profile will be blocked until the owner of the profile acknowledges these modifications by resetting the transfer admission to “valid”, for example with FTMODPRF <profile> TRANSFER-ADMISSION=*OLD-ADMISSION(VALID=*YES).

Creating admission profiles for foreign user IDs

When the FTAC administrator wants to create an admission profile for a foreign user by means of the FTCREPRF command, he can proceed in the following two ways:

– If the FTAC administrator possesses the SU privilege (see page 70), then he may set up admission profiles for other user IDs without restriction even if he does not know the current user password. The FTAC administrator may specify a TRANSFER-ADMISSION in these profiles. This can be used in FT requests immediately after being

set up. Please note that FTAC administrators who possess the "SU privilege" can get access to the files belonging to any and all user IDs by setting up the corresponding admission profiles and may therefore be able to by-pass protection mechanisms!

– Provided the FTAC administrator (without the SU privilege) knows all the data required for the USER-ADMISSION (i.e. user ID, account number and password) and specifies them when creating the admission profile, it is also possible to specify a TRANSFER-ADMISSION, with which a valid admission profile is created, i.e. the profile can immediately be used in file transfer and file management jobs.

  The password is stored as a part of this type of admission profile, so if a user changes his password, the admission profile also has to be changed.

  Example

  The FTAC administrator creates a valid admission profile for *USER1*. To do so, the administrator needs to enter the user's account number (*123456*) and password (*PASSWD1*).

```
FTCREPRF NAME=HISPROF1, TRANS-ADM=READYFORUSE, -
USER-ADM=(USER1,123456,PASSWD1)
```

– The FTAC administrator can also create an admission profile for a foreign user that does not contain the user's password. (When an FT job refers to this type of profile, FTAC enters the z/OS password currently valid for the user ID. That way the admission profile will not have to be changed should the z/OS password ever be modified.)

  In this case, the FTAC administrator (without the SU privilege) cannot specify TRANSFER-ADMISSION when creating the admission profile. That would create a locked admission profile, i.e. the profile can only be used in file transfer and file management jobs after the user has specified a TRANSFER-ADMISSION using the FTMODPRF command and after completed the USER-ADMISSION data.

  Example

  The FTAC administrator creates an admission profile for *USER1*. For the USER-ADMISSION, he specifies only the user ID, not the account number and the password. In that case the administrator may not specify a TRANSFER-ADMISSION.

```
FTCREPRF NAME=HISPROF2, TRANS-ADM=*NOT-SPECIFIED, -
USER-ADM=(USER1,*NOT-SPECIFIED,*NOT-SPECIFIED)
```

The FTAC administrator views the admission profile using the FTSHWPRF command. The short output shows that the profile is locked (indicated by the "!" in front of the profile name):

```
FTSHWPRF NAME=HISPROF2, SEL=(OWNER=*ALL)
  OWNER       NAME
  USER1      !HISPROF2
```

The long output shows that no valid TRANSFER-ADMISSION was specified in the profile:

```
FTSHWPRF NAME=HISPROF2, SEL=(OWNER=*ALL), INF=*ALL
 HISPROF2
  TRANS-ADM   = (NOT-SPECIFIED)
  USER-ADM    = (USER1,NOT-SPECIFIED,NOT-SPECIFIED)
  PROC-ADM    = SAME
  FT-FUNCTION = (TRANSFER-FILE, MODIFY-FILE-ATTRIBUTES,
                 READ-FILE-DIRECTORY)
  LAST-MODIF  = 2017-01-18 11:22:26
```

The user now assigns a TRANSFER-ADMISSION and supplements the USER-ADMISSION data:

```
FTMODPRF NAME=HISPROF2, TRANS-ADM=NOWREADYFORUSE, -
USER-ADM=(USER1,123456,PASSWD1)
```

Now the admission profile can be used in file transfer and file management jobs as well.

The user views the admission profile with the FTSHWPRF command.

The short output shows that the profile is no longer locked:

```
FTSHWPRF NAME=HISPROF2
  OWNER       NAME
  USER1       HISPROF2
```

The long output shows that the user's account number has been included in the admission profile along with the identifier YES for the USER-ADMISSION password:

```
FTSHWPRF NAME=HISPROF2, INF=*ALL
 HISPROF2
  USER-ADM    = (USER1,123456,YES)
  PROC-ADM    = SAME
  FT-FUNCTION = (TRANSFER-FILE, MODIFY-FILE-ATTRIBUTES,
                 READ-FILE-DIRECTORY)
  LAST-MODIF  = 2017-01-18 11:28:12
```

Privileging admission profiles

In exceptional cases, the FT user can use a privileged admission profile to disregard the specifications of own admission profile. Exceptional cases where this is allowed include:

- if a particular file needs to be transferred,
- if follow-up processing is not permitted or severely restricted,
- if a partner system with a higher security level is permitted to carry out file transfers with the user ID, but others with lower security levels are not.

The user ID protection is maintained in this case, by the fact that only very restricted access is permitted into the admission profile.

The procedure to follow when privileging an admission profile is simple:

1. The user creates an admission profile for the planned task with the command FTCREPRF.
2. The FTAC administrator views the admission profile with the command FTSHWPRF to determine if the profile presents a threat to data security.

   Example

   ```
   FTSHWPRF NAME=PROFPROD,
            SELECT-PARAMETER=(OWNER-IDENTIFICATION=STEVEN), -
            INFORMATION=*ALL
   ```

   Short form:

   ```
   FTSHWPRF PROFPROD,SEL=(,STEVEN),INF=*ALL
   ```

   The output has the following form:

   ```
   PROFPROD
    IGN-MAX-LEV = (IBR)
    FILE-NAME   = UMSATZ
    USER-ADM    = (STEFAN,M4711DON,OWN)
    PROC-ADM    = SAME
    SUCC-PROC   = NONE
    FAIL-PROC   = NONE
    FT-FUNCTION = (TRANSFER-FILE, MODIFY-FILE-ATTRIBUTES,
                      READ-FILE-DIRECTORY)
    LAST-MODIF   = 2017-01-18 11:43:57
   ```

   The first line of the output shows the name of the admission profile, the second line the values which STEVEN has set in the command FTCREPRF or which are determined by the default values, if Steven doesn't set them himself.
3. If the profile will not endanger security, the FTAC administrator privileges it with the help of the command MODIFY-FT-PROFILE.

Example

```
FTMODPRF NAME=PROFPROD,
         SELECT-PARAMETER=(OWNER-IDENTIFICATION=STEVEN), -
         PRIVILEGED=*YES
```

When used with the modified profile, the command FTSHWPRF UMSAWARE,SEL=(,STEFAN),INF=*ALL returns the same output as in the example above but with the addition of PRIVILEGED:

```
PROFPROD          PRIVILEGED
  IGN-MAX-LEV = (IBR)
  FILE-NAME     = UMSATZ
   ...
```

In a privileged admission profile, only the transfer admission and the parameter PRIVILEGED may be modified by the user. This prevents the misuse of any profiles, once privileged.

3.7.3 Saving and migrating the FTAC environment

When migrating individual users to another processor, or when migrating the complete processor, it is possible to provide the users with the same FTAC environment by saving the admission sets and FT profiles and restoring them on the new processor. Furthermore, you can also created backup copies of the FTAC environment on your processor by this method.

The following commands are available for migrating and saving the FTAC environment:

FTEXPENV	output FTAC environment to file
FTIMPENV	transfer FTAC environment from file
FTSHWENV	show FTAC environment from export file

Saving and importing admission sets and FT profiles

- When saving the FTAC environment you can select the admission sets and FT profiles which you wish to save for particular users. You must specify the name of the backup file.

  In all cases, the standard admission set is not included in the backup.

- When re-importing the saved FTAC environment you can make a distinction between sets, profiles and login names, i.e. you must not accept the entire backup contents. Please note that the values which refer to the standard admission set are always assigned to the values of the currently valid admission set.

  Depending on the rights of the FTAC administrator who is performing the import and the security settings in the "import system", it may be necessary to set up privileges explicitly on the new computer and release the transfer admissions explicitly.

- You can display the contents of a backup file with the command FTSHWENV.

Example

Steven Miller needs to work on a new computer under the same user ID STEVEN. Steven would like to keep the same admission set and admission profiles as before. To do this, the FTAC administrator Jack backs up the admission set and the admission profiles for the user ID STEVEN in the file STEVEN.FTAC.BKUP.

```
FTEXPENV TO-FILE=STEVEN.FTAC.BKUP,USER-IDENTIFICATION=STEVEN
```

Being a conscientious FTAC administrator, Jack checks if the desired backup 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   *PROFPROD
```

Now, Jack transfers the file STEVEN.FTAC.BKUP to the user ID of the FTAC administrator on the new computer.
There, Sylvester, the FTAC administrator for the new computer, transfers the admission set and the admission profiles of the user ID STEVEN from the file STEVEN.FTAC.BKUP.

Sylvester is also a conscientious administrator. He checks if Steven's admission set and profiles are a threat to the security of his system (he doesn't trust Jack in the slightest):

```
FTSHWENV FROM-FILE=STEVEN.FTAC.BKUP
```

and of course he receives the same output as above.

Then Sylvester imports Steven's admissions from the file STEVEN.FTAC.BKUP onto his system:

```
FTIMPENV FROM-FILE=STEVEN.FTAC.BKUP
```

Sylvester must then privilege Steven's profile:

```
FTMODPRF PROFPROD,,(,STEVEN),PRIV=*Y
```

Finally, Steven must release the imported profiles before he can work with them.

```
FTMODPRF NAME=*ALL,TRANSFER-ADMISSION=*OLD(VALID=*YES)
```

3.7.4 Administering the FTAC logging function

The FTAC logging function is integrated in the FT logging, i.e. you use the same commands, see section “FT logging” on page 140.

The display of FTAC logging records can not be turned off. However, the FTMODOPT command can be used to restrict it to requests rejected by FTAC (*REJECTED) or to modified requests (*MODIFICATIONS).

Beeing the FTAC administrator, you can use the FT command FTSHWLOG to find out about all access checks which have been carried out by openFT-AC to date. This facilitates processes such as system inspections.

The output of logging records and the meaning of the fields is explained in detail at the FTSHWLOG command, see manual "openFT (z/OS) - Command Interface".

3.8 Using openFT instances

In openFT you can run multiple openFT instances on one computer simultaneously. Because of these instances, should a computer fail, you are in a position, for example, to carry over the functionality of the openFT to another computer of a SYSPLEX configuration, which is already running openFT.

Following openFT installation, it is first necessary to use FJGEN to set up an instance. If you do not specify an instance name here then STD is used for the standard instance.

Up to 16 additional instances can be created by administration. Each of these instances, including the standard instance, consists of the following components:

- The request file SYSRQF, the partner list SYSPTF, the logging file SYSLOG, trace files, options SYSOPF and the profile file SYSFSA.
- Each instance requires its own network address; this always remains the same, independent of the real host.
  Therefore, the name of the host on which an instance is running is specified in the FJGEN command. This host name must always be accessible under the same network address.

The openFT installation files are only available once per computer and are shared by all the instances. The same version, however, must be installed on all the computers in the cluster (openFT version, release version, PTFs, etc.).

openFT commands that are called during a preprocessing, postprocessing or follow-up processing session, run under the same instance as the request that initiated the processing.

3.8.1 Setting up openFT instances

Instances are created by means of the FJGEN command (see page 56). They are identified and administered via the instance name that you specify with INSTANCE NAME in FJGEN. For the sake of clarity, the instance name should be a name part of all the openFT files and libraries that belong to the corresponding instance (e.g. FTAC files etc.).

WARNING!
The instance name should not be confused with the so-called instance identification that is defined using the IDENTIFICATION parameter in the FTMODOPT command. As of openFT V8.1, the instance identification is used by partner systems in order to authenticate your openFT instance. Similarly, you need these partner systems' instance identifications in order to authenticate them in the local system.

If you are only working with one instance then you should use the standard instance STD. This name is also proposed as the default in FJGEN.

Instance-specific CONN file

There is a so-called CONN file associated with each instance. It contains information required for internal communication between the command client from the library <openft qualifier>.OPENFT.NCLOAD and openFT from the library <openft qualifier>.OPENFT.LOAD and for encrypting this communication.

If you want to work with a specific instance then before you call any openFT functions, the instance-specific CONN file must be allocated by:

```
<openft qualifier>.<inst>.CONN
```

This is possible, for example, using the following call:

```
ALLOC DSNAME('<openft qualifier>.<inst>.CONN') DDNAME(OPENFT) SHR REUSE
```

Where <openft qualifier> and <inst> correspond to the OPENFT QUALIFIER and INSTANCE NAME specifications in the FJGEN command.

It is **urgently recommended** that you allocate the CONN file before calling the openFT command. This also applies if only the standard instance exists!

Instance-specific assignment of the NCLOAD

To allow openFT commands to be called under TSO or from a CLIST, the NCLOAD <openft qualifier>.OPENFT.NCLOAD must be entered in the search path/sequence for TSO commands. This can be done using the following command, for instance:

```
TSOLIB ACT DATASET(<openft qualifier>.OPENFT.NCLOAD)
```

Instance-specific CLIST

To administer openFT, it is also necessary to concatenate the instance-specific CLIST <openft qualifier>.<inst>.CLIST (either in the current TSO session or by incorporating it in the LOGON procedure, see page 46). This also applies to the standard instance.

If multiple openFT instances are to run in parallel on a computer under the same user ID, then different job names must be set in the FJBATCH members of the instance-specific CLISTS (for example, USERAX instead of USERAF). These are the batch jobs that load the appropriate openFT instances.

Exchange settings between instances

It is a simple matter to exchange partner entries between the instances using the LAYOUT=*ZOS-PROC parameter in the FTSHWPTN command (see the example for the FTSHWPTN command). FTAC components can be taken over using the commands FTEXPENV and FTIMPENV.

Show information about instances

You can use the FJGENPAR command to view the installation parameters of the current instance during operation (and modify them, if required, by means of a new FJGEN run). FTSHWINS allows you to obtain information on the known openFT instances running on a computer, provided that openFT has been started as a subsystem.

3.8.2 Importing an instance to another computer

The following steps are required to change over an openFT instance to another computer:

– Stop the instance on the original computer (FTSTOP).

– Unload the instance on the original computer (FTTERM). This unlocks all of the files required by openFT (request file, transfer files, etc.).

– Import the variable files, the network address and all of the files required by the requests to the destination computer. This can contain, among other things, the switching over of one or several pubsets).

– Load the instance on the destination computer (FJINIT).

– Start the instance on the destination computer (FTSTART).

After importing an instance to another computer, openFT finishes the (under some circumstances restartable) requests, whose admissions were already checked before importing. The new environment must have the same prerequisites as the old computer (the same IDs with the same file access admissions).

All file systems that are accessed by requests must be available. All requests whose file systems are not accessible during restart attempts are aborted.

On the new computer, the network view must be the same as that on the old computer. This means that the same host names for partner computers must be available and they must refer to the same partner computer. The network address of the host on which the instance is running, must be seen from the outside the same as from the address of the host, on which the instance was previously running.

The standard instance STD cannot be switched.

3.9 Backing up the configuration data

You should back up the configuration data of your openFT instance at regular intervals. This ensures that you will be able to restore openFT operation with as little delay as possible using the original runtime environment after a computer has failed or been replaced, for instance.

You should always store the operating parameter settings, the partner list and, where applicable, the FTAC environment in backup files. To do this, you can proceed as follows (the filenames are only examples and the backup files must not already exist):

- Backing up the operating parameter settings:

```
FREE DDNAME(SYSPRINT)

ALLOC DSNAME(OPTZOS.CLIST) DDNAME(SYSPRINT) NEW KEEP DSORG(PS) RECFM(F,B)
LRECL(80)

FTSHWOPT OUT=*STDOUT(*ZOS-PROC)

FREE DDNAME(SYSPRINT)
```

- Backing up partner list entries:

```
FREE DDNAME(SYSPRINT)

ALLOC DSNAME(PARTZOS.CLIST) DDNAME(SYSPRINT) NEW KEEP DSORG(PS) RECFM(F,B)
LRECL(80)

FTSHWPTN OUTPUT=*STDOUT(*ZOS-PROC)

FREE DDNAME(SYSPRINT)
```

- Backing up the FTAC environment:

```
FTEXPENV FTAC.SAVE
```

3.10 Controlling via an operator console

openFT can be controlled from an operator console. For administering an openFT instance via the operator console, the ID *Console* must be entered in the FTADM and, if necessary, the FTACADM members of the PARM parameter library. This is done by default during installation while the openFT instance is being generated with FJGEN.

Starting openFT via an operator console

openFT can be started from an operator console in the usual way using the START command (started task):

```
START openft-procname
```

In this case, openft-procname is the name of the start procedure for the started task. An example of such a start procedure is given in the section “openFT as a job or started task” on page 92.

3.10.1 Terminating openFT via an operator console

openFT can be terminated from an operator console using the STOP command. The STOP command is converted internally into a FTSTOP and FTTERM command.

Command format:

```
STOP openft-jobname
```

`openft-jobname` name of the openFT batch job or started task.

openFT can also be canceled from the operator console in the normal way using the CANCEL command. openFT does not convert this command internally.

3.10.2 Issuing administration commands via an operator console

You can also enter administration commands at an operator console in order to control openFT. The commands are entered as follows:

```
MODIFY openft-jobname, adm-command
```

or in abbreviated form:

```
F openft-jobname, adm-command
```

`openft-jobname`: name of the openFT batch job or started task.

`adm-command`: FT administration command.

All the FT administration commands described in the section “openFT commands for installation and initial operation” on page 104 and the manual "openFT (z/OS) - Command Interface" can be used here except FJGENPAR, FTHELP, FTSHWINS, FTSHWNET und FTTRACE.

If you enter the NCANCEL and NSTATUS commands at an operator console, they are interpreted as administration commands, i.e. you can use these commands to cancel and request information on all users' FT requests (privileged form of the command).

The FT administration commands must be entered as described in the manual "openFT (z/OS) - Command Interface".

Only one MODIFY command can be processed at any one time. If another MODIFY command occurs during processing, the following message is issued:

```
MODIFY REJECTED-TASK BUSY
```

In this case you must repeat the command.

The messages issued by openFT in response to the administration commands are displayed at the operator console at which the command was entered. Message lines which do not begin with an error code (e.g. lines in the FTSHWPTN output) are prefixed with the code FJM2000. Since messages are output via the WTO macro in "single line" format, output consisting of a number of lines may be interspersed with other system messages.

Example

You want to set the two parameters CONNECTION-LIMIT and PROCESS-LIMIT to the value 2 from an operator console. The name of the openFT batch job is USERAF.

You must enter the following command at the operator console:

```
MODIFY USERAF, FTMODOPT CONN-LIM=2, PROC-LIM=2
```

3.11 Controlling via NetView

openFT can be controlled via NetView or a NetView-compatible network management system (e.g. NetMaster). For it to be possible to administer an openFT instance via NetView, the ID *Console* must be entered in the FTADM and, if necessary, the FTACADM members of the PARM parameter library.

3.11.1 Starting openFT via NetView

openFT can be started under NetView as a started task. To do this, the system command START must be issued using the NetView command MVS:

```
MVS START openft-procname
```

In this case, openft-procname is the name of the start procedure for the started task. An example of such a start procedure is given in the section “openFT as a job or started task” on page 92.

3.11.2 Terminating openFT via NetView

You can also terminate openFT under NetView by issuing the STOP command as a system command. The STOP command is converted internally into an FTSTOP and FTTERM command.

Command format:

```
MVS STOP openft-jobname
```

`openft-jobname` name of the openFT batch job or started task.

In addition, the CANCEL command can be issued via NetView as a system command, thus canceling openFT. openFT does not convert this command internally.

3.11.3 Issuing administration commands via NetView

The MODIFY command can also be issued via NetView as a system command. Administration commands for controlling openFT can thus be entered as follows:

```
MVS MODIFY openft-jobname, adm-command
```

or in abbreviated form: `MVS F openft-jobname, adm-command`

`openft-jobname`: name of the openFT batch job or started task

`adm-command`: FT administration command

All the FT administration commands described in the section “openFT commands for installation and initial operation” on page 104 and the manual "openFT (z/OS) - Command Interface" can be used here except FJGENPAR, FTHELP, FTSHWINS, FTSHWNET und FTTRACE.

If you enter the NCANCEL and NSTATUS commands at an operator console, they are interpreted as administration commands, i.e. you can use these commands to cancel and request information on all users' FT requests (privileged form of the command).

The FT administration commands must be entered as described in the manual "openFT (z/OS) - Command Interface".

Only one MODIFY command can be processed at any one time. If another MODIFY command occurs during processing, the following message is issued:

```
MODIFY REJECTED-TASK BUSY
```

In this case you must repeat the command.

The messages issued by openFT in response to the administration commands are sent to the NetView console at which the command was entered. Message lines which do not begin with an error code (e.g. lines in the FTSHWPTN output) are prefixed with the code FJM2000. The messages can then be processed using NetView-specific functions. Since messages are output via the WTO macro in "single line" format, output consisting of a number of lines may be interspersed with other system messages.

Example

You want to set the two parameters CONNECTION-LIMIT and PROCESS-LIMIT on the value 2 under NetView. In addition, the partner HOSTA is to be deactivated. The name of the openFT batch job is USERAF. You must enter the following commands at the NetView console one after the other:

```
MVS MODIFY USERAF, FTMODOPT CONN-LIM=2,PROC-LIM=2
MVS MODIFY USERAF, FTMODPTN HOSTA,STATE=*DEACT
```

4 Remote administration

4.1 Configuring an openFT instance on the z/OS system for remote administration

The remote administration server uses FTAC transfer admissions to access the openFT instances. This means that the appropriate admission profiles must be defined in the openFT instances from which administration is being carried out.

To enable a remote administrator to access the openFT instance, the FT administrator creates an admission profile on the z/OS system using the REMOTE-ADMINISTRATION function:

```
FTCREPRF NAME=profile                                  –
        ,TRANSFER-ADMISSION=transfer_admission         –
        ,PARTNER=remote_administration_server          –
        ,FT-FUNCTION=*REMOTE-ADMINISTRATION
```

The ADM administrator specifies the FTAC transfer admission in the configuration file of the remote administration server when defining the openFT instance. For an example, see the manual "openFT (Unix and Windows systems) - Installation and Operation". The operand PARTNER= ensures that this profile can only be used by the remote administration server.

Entering the remote administration server in the partner list

If remote administration requests are to be issued from your z/OS system, the FT administrator can enter the remote administration server in the partner list. This has the advantage that you can explicitly assign particular attributes to this partner, for instance the security level or the trace settings.

The FT administrator enters the remote administration server in the partner list using the following format:

```
ftadm://host[:port number]
```

You only specify *port number* if the default ADM port (11000) is not used on the remote administration server *host*. The same applies if a remote administrator specifies the address directly in a remote administration request.

4.2 Issuing remote administration requests

If you wish to enter remote administration requests, you require the following:

- the name of the remote administration server in the partner list or the address of the remote administration server (ask the FT administrator if necessary)
- the transfer admission for accessing the remote administration server. The ADM administrator of the remote administration server must make this available to you.

You are able to determine the names of the openFT instances that you are permitted to administer yourself.

Determining the names of the openFT instances

The ADM administrator defines the names of the openFT instances during configuration of the remote administration server. You get the names of the openFT instances by executing the ftshwc command as a remote administration command on the remote administration server:

```
FTADM PARTNER-SERVER=server                        –
     ,TRANSFER-ADMISSION=transfer_admission        –
     ,ROUTING-INFO=*NONE                           –
     ,CMD='ftshwc -rt=i'
```

Explanation

server
: Name of the remote administration server from the partner list. Alternatively, you can also enter the address directly in the format *ftadm://host...*

transfer_admission
: FTAC transfer admission on the remote administration server.

'ftshwc -rt=i'
: 'ftshwc -rt=i' is a command executed on the remote administration server that outputs the names of the instances that you are permitted to administer. You must enter the quotes.

Sample output

```
TYPE    = *INSTANCE     ACCESS = FT+FTOP          MODE = FTADM
   NAME = Muenchen/MCH1/OPENFT01
   DESC = Windows Server 2012
TYPE    = *INSTANCE     ACCESS = FT+FTOP          MODE = FTADM
   NAME = Muenchen/MCH1/OPENFT02
   DESC = Solaris
TYPE    = *INSTANCE     ACCESS = FTOP             MODE = LEGACY
```

```
   NAME = Muenchen/MCH1/OPENFT03
   DESC = Windows Server 2016
TYPE    = *INSTANCE     ACCESS = FT+FTOP+FTAC     MODE = FTADM
   NAME = Muenchen/MCH2/MCHSRV03
```

NAME specifies the name of the instance that you must specify exactly as given here in the remote administration request., see also command FTADM in the manual "openFT (z/OS) - Command Interface". Your remote administration permissions for this instance are listed under ACCESS. MODE specifies whether the instance is administered via the FTADM protocol (MODE=FTADM) or via ftexec (MODE=LEGACY).

Issuing a remote administration request

Specify the remote administration command in the following form:

```
FTADM PARTNER-SERVER=server                        -
     ,TRANSFER-ADMISSION=transfer_admission        -
     ,ROUTING-INFO=instance                        -
     ,CMD='command'
```

Explanation

server
: Name of the remote administration server from the partner list. Alternatively, you can also enter the address directly in the format *ftadm://host...*

transfer_admission
: FTAC transfer admission on the remote administration server.

instance
: Routing name of the openFT instance on which the administration command is to be executed. You must enter this name in exactly the form in which it appears on the remote administration server with the ftshwc command. See “Determining the names of the openFT instances” on page 166.

command
: Specifies the administration command to be executed on the openFT instance. For further details, see command FTADM in the manual "openFT (z/OS) - Command Interface".

4.3 Logging remote administration

ADM log records are created in each of the openFT instances involved when remote administration requests are issued.

ADM log records are explicitly flagged as being of a particular type (A). They are handled in a similar way to FT or FTAC log records, i.e. you can view ADM log records in z/OS system using the FTSHWLOG command and delete them with the FTDELLOG command (provided that you have the appropriate permission to do so). Please refer to the manual "openFT (z/OS) - Command Interface" for details.

Controlling ADM logging

The FT administrator controls the scope of ADM logging using the operating parameters. The following options are available:

- log all administration requests
- log all administration requests that modify data
- log administration requests during which errors occurred
- disable ADM logging

You do this by means of the FTMODOPT command with the operand LOGGING=*SELECT(ADM=...)

4.4 ADM traps

4.4.1 Configuring ADM traps in the openFT instance

To allow ADM traps from your openFT instance on the z/OS system to be sent to the ADM trap server, you must carry out the following actions in your role as FT administrator:

- Enter the address and admission data for the ADM trap server
- Specify the scope of the ADM traps sent to the ADM trap server

In addition, the FT administrator of the ADM trap server must set up a corresponding admission profile on the ADM trap server.

Enter the address and admission data for the ADM trap server

You specify the address and the transfer admission of the ADM trap server in the ADM-TRAPS operand of the FTMODOPT command:

```
FTMODOPT ...                                                    –
         ,ADM-TRAPS=*PAR(DESTINATION=(PARTNER=adm-trap-server,  –
                     TRANSFER-ADMISSION=trap-admission))
```

adm-trap-server
: must be defined in the partner list using the address format *ftadm://host...*. Alternatively, you can also enter the address directly in the format *ftadm://host...*

trap-admission
: is the transfer admission for the admission profile defined in the ADM trap server for this purpose.

Specify the scope of the ADM traps

The scope of the ADM traps sent to the ADM trap server is controlled using the operating parameters. You can set which of the events listed below cause traps to be sent:

- Change of openFT status (FTSTART / FTSTOP)
- Change of partner status
- Unavailability of partners
- Change of request management status
- Successfully completed requests
- Failed requests

To do this, use the FTMODOPT command and defying the required selection under SELECTION in the ADM-TRAPS operand.

4.4.2 Viewing ADM traps

The FT administrator of the ADM trap server is permitted to view all ADM traps on the ADM trap server. If the ADM trap server is also used as the remote administration server, the remote administrators can also view traps.

If you log on to your z/OS system as a remote administrator, you can view your "own" ADM traps. These are the ADM traps of those openFT instances for which you have at least FTOP permission. See the “Determining the names of the openFT instances” on page 166.

If you wish to view the most recent 10 ADM traps, enter the following remote administration command:

```
FTADM PARTNER-SERVER=server                    -
     ,TRANSFER-ADMISSION=transfer_admission    -
     ,ROUTING-INFO=*NONE                       -
     ,CMD='ftshwatp -nb=10'
```

Explanation

server
: Name of the remote administration server from the partner list. Alternatively, you can also enter the address directly in the format *ftadm://host...*

transfer_admission
: FTAC transfer admission on the remote administration server.

'ftshwatp -nb=10'
: 'ftshwatp -nb=10' is a command executed on the remote administration server that outputs the last 10 ADM traps. You must enter the quotes.

The ftshwatp command also provides further options. For details, see, for instance, the manual "openFT (Unix and Windows systems) - Command Interface".

5 Diagnostics

This chapter contains the following sections:

5.1 General notes

The measures to be taken when an error message occurs are described under the message involved (see page 199).

The tips given in section “Optimizing the operating parameters” on page 120 should help you rectify faults or bottlenecks which occur during FT operation.

If serious errors occur that lead to openFT terminating with a dump, the following information may be of help when trying to find the reason for the error: If an user abend code with a value below 4094 is reported, the code corresponds to a system abend code in decimal presentation (e.g. user abend code 1667 equals system abend code 683).

As an FT administrator, you must also advise FT users who are in doubt or who cannot rectify certain errors themselves. The section "Hints for the FT user" in the manual "openFT (z/OS) - Command Interface" can help you in this case.

If, despite taking every precaution, an error occurs that neither you nor the system administrator can resolve, please create diagnostic records as described in Creating diagnostic records, and contact your Service Center.

You can also use the FTPING command to test the response from a remote partner. The interface of FTPING on z/OS is exactly similar to the FTPING syntax on Unix and Windows platforms. The syntax is displayed when you enter FTPING without options. FTPING will be available to the user only on TSO environment.

5.2 Creating diagnostic records

For diagnostics, the following traces and information features are available:

Traces

openFT	FT-Trace	supplies information about the NEABF protocol.
GTF trace	including SVC Trace	supplies information about the progress of SVC trace execution.
VTAM	buffer trace	supplies information about data traffic between VTAM and VTAM application.
VTAM	line trace	supplies information about data traffic via the line (SDLC protocol).

Obtaining information

openFT	NSTATUS command indicates the status of requests.
VTAM	DISPLAY command indicates status of local LUs and PUs
NETVIEW	indicates the status of the network.
NETSTAT	provides information about the TCP/IP network.

If, despite due care and attention, an error occurs that neither the FT administrator nor the z/OS system administrator can rectify, contact your Service Center. To facilitate trouble-shooting, please submit the following:

- detailed description of the error situation and statement indicating whether the error is reproducible;
- openFT trace files see page 175

  If possible, the trace files should be formatted with the FTTRACE command, and, if applicable, the FT trace from the remote FT system. Run trace to cover a longer period (>= 2 h) in order to provide sufficient time stamps and possibly connection cleardown or, in the case of reproducible errors, activate trace **before** reproducing the same error).
- if applicable the command call and result list of the request that triggered the error
- job list of the openFT job (also from partner system is possible)

- general information as for z/OS system error:
  - type of system (z/OS,...) and system version
  - name and version of the job entry subsystem installed (JES2, JES3, ...)
  - information about the data protection support installed with name and version (SYS1.UADS, RACF, TOP-SECRET, ACF-2, MVS router exit, openFT-AC)
  - version of the Data Facility Product (DFP) installed, if applicable
  - openFT version installed
  - complete list of openFT corrections used
- version of the FT partner and details of the transport system (e.g. DCAM, CCP / CMX, VTAM, etc.)
- openFT dump files with the name '<openft qualifier>.<inst>.SYSFDF.Dyymmdd.Thhmmss' or the SYSUDUMP assigned in the FJBATCH- job
- If necessary, create an HPNS trace if problems arise with respect to TCP/IP. To do this, you must create a member DIAGPAR in the instance-specific PARM library and either restart the batch job or issue the FTUPDADDR command. See DIAGPAR member in the FT parameter library. If the diagnostic event does not itself generate an openFT dump (<inst>.SYSUDUMP) together with the associated HPNS trace then the openFT batch job must be canceled with dump output.

The output from the FTINFO command can also be of use. This only functions for inbound requests and must therefore by called at TSO level using the FTEXEC command. The partner name in this case is the local host:

```
FTEXEC HOSTNAME,'ftinfo -csv',(<userId>,<account>,<password>)
```

would output the following:

```
CmdUiVer;CmdTiVer;OsType;UserId;IsFtAdm;IsFtacAdm;FtLang;CcsName;Home;Limited;
IsAdmAdm;ProdVer;SrcVer;Inst;TimeOffset;FtScriptDir;NativeX25;SingleUser;Crypt

1210;0;"z/OS";"OPFTWIT";1;1;"en";"IBM1047";"OPFTWIT";*NO;0;"12.1A00";"354";"STD";
3600;"";*NO;*NO;*YES
```

DIAGPAR member in the FT parameter library

To allow additional diagnosis with an HPNS trace, the member DIAGPAR must be supplied with the following values in the instance-specific FT parameter library PARM:

HPNSTRACE=17825791

DIAGSTAMPS=12

If the openFT (batch job) is restarted or the FTUPDADDR command is called with these entries, additional diagnosis records on data communication are returned at the socket interface or written to the file <inst>.SYSUDUMP in the event of a dump written to the file <inst>.SYSUDUMP.

If you restart the FJBATCH job – irrespective of whether or not a dump has previously been written –, then the dump file is renamed to <inst>.SYSUDUMP.PREV. As a result, it is not deleted immediately and can be used for diagnostic purposes. This only functions if the FJBATCH job is constructed as depicted on page 93.

5.3 Trace function

5.3.1 Controlling the trace function

The FT administrator uses the following commands to control the trace function:

FTADDPTN	Add a remote system to the partner list
FTMODOPT	Modify operating parameters
FTMODPTN	Modify partner properties in the partner list

The FT administrator uses the following commands to get information on the current settings:

FTSHWOPT	Information about operating parameters
FTSHWPTN	Information about partner systems

The FT trace function can be switched on and off irrespective of whether the FT system is active or inactive.

You can set the scope of openFT traces globally using the FTMODOPT command. You can differentiate by partner type (openFT, FTP), request type (local/remote and synchronous/asynchronous) and trace scope (with/without file contents). The global setting can be modified on a partner-specific basis using FTMODPTN (or set before with FTADDPTN).

The following table illustrates four typical cases of trace use.

FTMODOPT	**FTMODPTN / FTADDPTN**	**Task**	**Effect**
TRACE=*ON	TRACE= *BY-FT-OPTIONS	General tracing of FT operations.	FT operation is fully traced.
TRACE=(SWITCH=ON, OPTIONS= NO-BULK-DATA)	TRACE= *BY-FT-OPTIONS	Connect tracing for all openFT partners.	Mass data transfers are not recorded. Recommended for long-lived traces.
TRACE=(SWITCH=ON ,PART-SELECTION= *FTP)	TRACE= *BY-FT-OPTIONS	Tracing of a a certain type of partner over an extended period. (here, ftp partners)	All events relating to a selected partner type are logged. Despite the extended period, the trace volume does not become excessive.

FTMODOPT	FTMODPTN / FTADDPTN	Task	Effect
TRACE=(SWITCH=ON ,REQ-SELECTION= *REM)	TRACE= *BY-FT-OPTIONS	Tracing of a specific type of request (here, requests submitted by a remote system)	All events relating to certain request types are logged. Despite the extended period, the trace volume does not become excessive.

The default value for FTADDPTN is BY-FT-OPTIONS. The global settings are thus taken over from FTMODOPT.

The following table indicates the interrelations between the most important FTMODOPT and FTADDPTN/FTMODPTN trace settings.

FTMODOPT	FTMODPTN / FTADDPTN	Effect
TRACE=*OFF	equals	*OFF
TRACE=*ON	TRACE=*BY-FT-OPTIONS	*ON
	TRACE=*UNCHANGED	Setting retained
	TRACE=*ON	*ON
	TRACE=*OFF	*OFF
TRACE=(SWITCH=ON, PARTNER-SELECTION= partner type)	TRACE=*BY-FT-OPTIONS	*ON if suitable partner type *OFF if unsuitable partner type
	TRACE=*UNCHANGED	Setting retained
	TRACE=*ON	*ON
	TRACE=*OFF	*OFF
TRACE=(SWITCH=ON, REQUEST-SELECTION= request type)	TRACE=*BY-FT-OPTIONS	*ON if suitable request type *OFF if unsuitable request type
	TRACE=*UNCHANGED	Setting retained
	TRACE=*ON	as *BY-FT-OPTIONS
	TRACE=*OFF	*OFF

You will find details on the trace files and the way they are formatted with FTTRACE in the section “Format of the trace files” on page 177 and in the section “FTTRACE command” on page 178.

5.3.2 FTTRACE - Convert trace data to readable form

Trace data which has been generated using the trace function (see FTMODOPT command, TRACE operand) can be converted into a readable form using the FTTRACE command. Before issuing this command, you must deactivate the trace function. This command can only be entered in TSO command mode.

5.3.2.1 Format of the trace files

openFT writes trace data to files with the following format:

– '<openft qualifier>.<inst>.Smddhhmm.Ssscc.I000.FTTF'
(Control process)

– '<openft qualifier>.<inst>.Smddhhmm.Ssscc.Iiii.FTTF'
(Server process for inbound and asynchronous outbound requests, i= 001,002, ...)

– '<openft qualifier>.<inst>.Ymddhhmm.Ssscc.Pnnnnnnn.FTTF'
(Process for synchronous outbound requests)

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

mddhhmm.Ssscc specifies the creation time of the trace file. Here, m indicates the month (1 = January, 2 = February, ... A= October, B=November, C = December), dd the day, hhmm the time in hours (hh) and minutes (mm), ssccc the time in seconds (ss) and milliseconds (ccc). nnnnnnn means the process ID of the process for synchronous outbound requests.

Please note that the trace file name may be shortened if the OPENFT QUALIFIER does not consist solely of a "first level qualifier", i.e. it contains a period. For example, ssccc may replaced by sss or may be omitted completely.

Trace files in the event of errors

– If a trace file cannot be written without errors due to a memory bottleneck, a DLOG record and a console message are output.

– If a record of the trace file cannot be written as a result of an infringement of the maximum record length, the trace file is closed and the subsequent records are written to a new continuation file with the additional suffix.Liii, e.g.:
’<openft qualifier>.<inst>.S8101010.S33222.I001.FTTF’ (first trace file)
’<openft qualifier>.<inst>.S8101010.S33222.I001.L001.FTTF’ (continuation file)

5.3.2.2 FTTRACE command

The FTTRACE command writes the converted trace to SYSPRINT, which is normally directed to the TSO console.
To write the converted trace to a file, the file must first be allocated, e.g.:

– FREE DDNAME(SYSPRINT)
– ALLOC DSNAME(TEST.TRACOUT) DDNAME(SYSPRINT) NEW CATALOG
– FTTRACE STD.S3141220.S44944.P3473434.FTTF
– FREE DDNAME(SYSPRINT)
– ALLOC DSNAME(*) DDNAME(SYSPRINT)

For large traces, sufficient storage space must be provided in the ALLOC command using the SPACE parameter.

Format

The FTTRACE command was originally implemented on the open platforms and then ported to z/OS.

```
fttrace  -h |
         [ -d ]
         [ -sl=n | -sl=l | -sl=m | -sl=h ]
         [ -cxid=<context id> ]
         [ -f=hh:mm:ss ]
         [ -t=hh:mm:ss ]
           <trace files>
```

Description

-h Outputs the command syntax on screen. Any specifications after *-h* are ignored.

-d Specifies that the trace files are to be output in hexadecimal format (dump format).

If you do not specify *-d* then the files are output in printable form, default value.

CAUTION!
In dump format, data which is relevant to security is also output in unencrypted form. Specifying a security level (-sl) is meaningless here.

-sl=n | **-sl=l** | **-sl=m** | **-sl=h**
Specifies the security level for the output.

n (no) No security requirements, i.e. all data is output including IDs, transfer admissions, passwords, file names etc.

l (low) Passwords are overwritten with XXX.

m (medium)
Passwords, user IDs, transfer admissions, account numbers, and follow-up processing commands are overwritten with XXX, default value.

h (high)
Passwords, user IDs, transfer admissions, account numbers, follow-up processing commands and file names are overwritten with XXX, default value.

-cxid=context id
Selects the trace entries on the basis of the context ID. This is made up as follows: the first character is the slot pool ID and the second to fourth characters are the ID of the slot. If you omit *-cxid* or specify *-cxid=* without a context ID then trace entries are output for all context IDs.

-f=hh:mm:ss (from)
Specifies the time as of which trace entries in the trace file are to be evaluated. You enter the time in the format hours:minutes:seconds (2 digits each).
If you do not specify a start time then trace entries are output from the start of the file.

-t=hh:mm:ss (to)
Specifies the time up to which trace entries in the trace file are to be evaluated. You enter the time in the format hours:minutes:seconds (2 digits each).
If you do not specify an end time then trace entries are output up to the end of the file.

trace files
Name(s) of the trace file(s) that you want to evaluate. You can specify multiple trace files and wildcards can be used.

Example

As an FT administrator, you want to create a trace file and then convert the data contained in this file into a readable form. You must take the following steps:

– switch on the trace function (in administration mode),
– switch off the trace function (in administration mode),
– convert the trace data into a readable form (in TSO command mode).

The commands you must enter are shown below:

```
READY
ftmodopt trace=*on
READY
.
   .                            (period during which the trace data
   .                             is being logged)
   .
READY
ftmodopt trace=*off
READY
fttrace std.S4051730.S13145.P1234567.FTTF
   (Trace data is output to screen)
READY
```

By default, FTTRACE outputs the data to the TSO console. If the data is to be output to file, you must allocate SYSPRINT accordingly before FTTRACE is called.

5.4 Rectifying problems with the OPFT subsystem

The OPFT subsystem is used for secure encryption of the user command and to provide the "right" user ID for later authentication (RACF etc.).

Components of the subsystem

The subsystem consists of three components (load modules) which must be assigned to the LPALIB when the computer is started (IPL):

– OPFTINIT
– OPFTSUB
– IGX00nnn, where IGX00211 is permanently predefined.

If IGX00211 is already assigned, another module with a number nnn from the range 200 … 255 with the basis OPFTIGX must be linked (see LINK skeleton).

Following IPL and the first startup of openFT, the status of the subsystem can be displayed using the following system command:

```
D SSI,SUB=OPFT
```

The problems described below can mainly be attributed to inadequate generation or interpretation and are thus referred to as "generation error type n" (n=1 through 5).

Generation error type 1

After openFT has been started, the following error messages are output one after the other:

```
FTR4199 OPENFT: SYSTEM ERROR. ERRORCODE ADM: 3410,nnnn
```

```
FTR4121 OPENFT: TERMINATED.
```

However, **no** message of the type OPENFT: SUBSYSTEM... is displayed on the system console.

Cause and possible solution

A subsystem status output further localizes the problem:

– If no information on the OPFT subsystem is available, the LPALIB is not or is not fully equipped with the modules listed under "Components of the subsystem".

– If the subsystem is of the type "static", it was created explicitly in the generation.

– If the subsystem does not have the attribute COMMANDS=REJECT, it was also created in the generation.

Omitting these specifications concerning the subsystem and the SVC in the generation solves the problem. An IPL is required.

The return value nnnn from the subsystem in the message FTR4199 is only informative to a certain degree.

Generation error type 2

As with type 1, messages FTR4199 and FTR4121 are issued. In contrast to type 1, the following three messages are displayed on the system console:

```
IEW4000I FETCH FOR MODULE OPFTSUB FROM DDNAME STEPLIB FAILED BECAUSE
INSUFFICIENT STORAGE AVAILABLE

OPENFT: SUBSYSTEM CREATION FAILED

OPENFT: RC / REASON 00000016 / 00000000
```

Cause and possible solution

With this problem, insufficient space is available in the common service area (CSA) to load the subsystem.

Increasing the size of the CSA pool in the generation solves the problem. A renewed IPL is required.

If message IEW4000I appears when operating openFT without a subsstem, the region specification in the batch job must be extended.

Generation error type 3

Following successful operation of openFT, the session is terminated by FTTERM and then restarted. Subsequently messages FTR4199 and FTR4122 are issued again, but no system messages are displayed on the console.

The subsystem status query shows that it is inactive.

Cause and possible solution

In this case the modules of the subsystem were placed not in the LPALIB but in the LINKLIB, which then led to the subsystem being "unloaded" when FTTERM took place.

Taking the modules over into the LPALIB when the next IPL takes place solves the problem.

Generation error type 4

The dialog connection to openFT is rejected with one of the following error messages:

```
FTR4193 OPENFT: OPENFT NOT AVAILABLE
```

```
FTR4196 OPENFT: DIALOG TASK VERSION INCOMPATIBLE
```

At the same time the CONN file in the connection to the client was dispensed with.

Cause and possible solution

In this case no standard instance STD was generated, but such an instance is necessary for a connection without a CONN file.

After the standard instance has been generated, a dialog connection can be established to this instance without a CONN file.

This error is also to be expected when working with the standard instance STD if an SVC number other than 211 (default) was used.

Generation error type 5

Starting openFT leads immediately to abortion with a dump.

Cause and possible solution

In this case the modules of the subsystem were copied from the openFT LOADLIB to the LPALIB without at least deleting OPFTSUB in the LOADLIB. As a result the system - if the LOADLIB is apf-authorized - takes OPFTSUB from the LOADLIB and not from the LPA. This leads to address error 0C4.

After OPFTSUB has been deleted in the LOADLIB, a new IPL must be created.

5.5 Additional diagnostic information

5.5.1 FJVERS - Display openFT load module versions

The FJVERS command is used to display the versions of the installed openFT load modules. This command can only be entered in TSO command mode.

FJVERS

Example

If openFT-AC and openFT-FTP are installed, the output may look like this:

```
 READY
fjvers
 VERSION OF 'OPENFTAC' IN LIBRARY,,'OPFTCHS.OPENFT.LOAD' IS '12.1A00 FTAC'
 VERSION OF 'OPENFTP' IN LIBRARY,,'OPFTCHS.OPENFT.LOAD' IS '12.1A00'
 VERSION OF 'OPENFT' IN LIBRARY,,'OPFTCHS.OPENFT.LOAD' IS '12.1A00'
 VERSION OF 'FTTRACE' IN LIBRARY,,'OPFTCHS.OPENFT.NCLOAD' IS '12.1A00,
 VERSION OF 'NCOPY' IN LIBRARY,,'OPFTCHS.OPENFT.NCLOAD' IS '12.1A00'
 READY
```

5.5.2 FTSHWD - Display diagnostic information

The FTSHWD command outputs any diagnostic codes (together with date and time) that may have been written during the error event.

FTSHWD

Without operands

Example

FTSHWD supplies the following output:

```
FTSHWD
DATE          TIME      SSID  COMPONENT    LOCATION-ID        INFO
20170125      131251    FT    79/yfasdia   3/EuisyMsg         fd00000c
```

6 Appendix

This chapter contains the following sections:

6.1 Accounting records

Structure of openFT accounting records

An openFT accounting record is divided into the following parts:

– SMF header
– record definition
– product information
– FT administrator area
– user information
– basic information
– file information

The following description of these record sections includes the absolute and relative **offsets** (relative to the start of the SMF record or the start of the record section being described), the **length** (in bytes) and the **format** of the data field. The following abbreviations are used when specifying the formats:

A alphanumeric

B binary

C printable character

F file name for z/OS

P packed decimal number

Z unpacked decimal number

Layout of the SMF header

Offsets (hex.)		Length (dec.)	Format	Description
abs	rel			
00	00	2	B	Length of the SMF record (including the length field) (1)
02	02	2	B	Segment descriptor (1)
04	04	1	B	System indicator "0xxxxx10" = OS/VS2 (2)
05	05	1	B	Record type (128, ..., 255)
06	06	4	B	Record storage time in hundredths of seconds since 0:00 local time
0A	0A	4	P	Record storage date in the format 0CYYDDDF (3)
0E	0E	4	C	System ID (from the SID parameter)

(1) The fields "length of the SMF record" and "segment descriptor" together form the record descriptor word (RDW). Depending on the reading method used, the RDW may be missing from the SMF records read out. The segment descriptor is set to "0000", i.e. only non-spanned records are written.

(2) Bits specified with "x" are reserved by IBM and are set by SMF under certain circumstances.

(3)
C : centuries later than the 20th century
YY : year
DDD: days in year
F : sign (= X'F')

Layout of the record definition section

Offsets (hex.) abs	Offsets (hex.) rel	Length (dec.)	Format	Description
12	00	4	A	Record ID ("FTR0")
16	04	2	C	Record version ("1A") (4)
18	06	2	B	Offset for product information (5)
1A	08	2	B	Offset for FT administrator area (5)
1C	0A	2	B	Offset for user information (5)
1E	0C	2	B	Offset for basic information (5)
20	0E	2	B	Offset for file information (5)

(4) An analysis program can recognize the structure of the accounting record from the record version. The structure described here corresponds to version "1A"; accounting records with a different structure (subsequent versions) are identified where appropriate by the corresponding record versions. The following is guaranteed for record versions "1A", "1B" etc.:

- The order in which the offset information is described here is retained in the record definition section.
- The structure of the record sections described here (product information, FT administrator area, etc.) is retained; if necessary, additional information is appended at the end of the relevant record section.

(5) These offsets are given in relation to the start of the SMF record. If, after the SMF record has been read, the record descriptor word (see above) is missing, 4 bytes must be subtracted from the specified offsets.

Layout of the product information

Offsets (hex.)		Length (dec.)	Format	Description
abs	rel			
22	00	6	C	Product name ("openFT")
28	06	4	C	Product version ("120A0")

Layout of the FT administrator area

Offsets (hex.)		Length (dec.)	Format	Description
abs	rel			
2C	00	40	C	FT administrator area (data from SMF_ADM_AREA; see the section “Setting up the FT parameter library” on page 57)

Layout of the user information

Offsets (hex.)		Length (dec.)	Format	Description
abs	rel			
54	00	8	A	User ID from the TRANSFER-ADMISSION
5C	08	40	C	"accounting information" from the TRANSFER-ADMISSION
84	30	8	A	User ID of the user who submitted the request (only for transfer requests submitted in the local system)

Layout of the basic information

Offsets (hex.) abs	Offsets (hex.) rel	Length (dec.)	Format	Description
8C	00	12	Z	Time when the file transfer request was stored, in the format YYMMDDhhmmss (applies only to requests issued in the local system)
98	0C	12	Z	Time when the transfer ended, format YYMMDDhhmmss
A4	18	1	C	Result of the transfer: + : successful transfer, - : unsuccessful transfer
A5	19	1	C	Follow-up processing in the local system + : was started - : was not started 0 : was not specified
A6	1A	8	A	Name of the remote system
AE	22	1	A	Transfer request was submitted L : in the local system, R : in the remote system
AF	23	11	Z	Transfer ID
BA	2E	2	-	Reserved
BC	30	4	B	Number of disk accesses (6)
C0	34	4	B	Number of bytes on disk (7)
C4	38	4	B	Number of bytes in network (8)

(6) Restrictions:
- For VSAM files, the number of control intervals is specified instead of the number of times the disk is accessed.
- When PO members are written with *EXTEND, a copy is first made of the old member. This counts as 1 disk access only.
- If restarts take place during file transfer, slight inaccuracies may occur in determining the number of times the disk is accessed.

(7) For files with record format V (or, in the case of VSAM files, where MAXLRECL is not equal to the AVGLRECL), the number of data bytes plus four times the number of records is specified. For VSAM files, the bytes reserved for the control interval definition fields and record definition fields are not taken into account.

(8) In the openFT protocol approximately the number of data bytes plus six times the number of records for variable record length files, else (fixed or undiefined record length) the number of data bytes. In practise, this value will slightly differ due to protocol data exchange, tabulator expansion, code conversion and so on. The use of data compression normally leads to lower values.

Layout of the file information

Offsets (hex.)		Length (dec.)	Format	Description
abs	rel			
C8	00	2	B	Length of the file name
CA	02	2	-	Reserved
CC	04	See below	F	File name (9)

(9) The length of this data field is specified in the data field "length of the file name" (maximum length: 56).

6.2 The openFT job log

The openFT job log contains the following information:

- z/OS messages caused by openFT, e.g.:

```
IEC130I OPFTPARM DD STATEMENT MISSING
IEF212I ........ ...... OPFTPARM – DATA SET NOT FOUND
IEF212I ........ ...... DDUADS – DATA SET NOT FOUND
```

(see the description of the FJGEN command, page 104).

```
CSV003I REQUESTED MODULE OPENFTCR NOT FOUND
```

(see section “Installation of the openFT-CR delivery unit” on page 48)

Which of these system messages are actually displayed also depends on your system environment.

- FT administration commands and the associated openFT synchronous messages:
  - FT administration commands which were entered at an operator console (possibly under NetView); these are indicated by an arrow "===>", e.g.:

    ```
    ===> FTSHWOPT
    ```

  - FT administration commands which were entered from TSO sessions; these are indicated by an arrow in the form "+++>", e.g.:

    ```
    +++>  FTSTART************
    FTR0500 OPENFT: openFT 12.1A00 starting. Protocols: openFT,FTP,ADM
    ```

  In the case of FT administrator commands whose names begin with FT, no parameters are output in the openFT job log, regardless of how many parameters were entered, e.g. the command FTSHWLOG is displayed as follows:

  ```
  +++> FTSHWLOG ************
  ```

  or

  ```
  ===> FTSHWLOG ************
  ```

  Synchronous messages issued by openFT in response to FT administration commands which were entered at an operator console (possibly under NetView) also are output by z/OS at the start of the job log, preceded by a plus sign "+".

- Asynchronous openFT messages:

  If asynchronous messages occur, they are always recorded in the openFT job log, regardless of whether

  – they are output at a TSO terminal at which FT administration mode is switched on (or they are collected for output at this type of terminal), or

  – they are output additionally or exclusively at one or more consoles; in this case, they also appear at the start of the job log, preceded by a plus sign "+".

- RACF messages concerning the rejection of checks in the context of transfer requests.

- Asynchronous messages output to one or more consoles on successful/unsuccessful file transfer (see keywords SUCC_MSG, FAIL_MSG and ENDMSG_ROUTCDE in the PARM member of the FT parameter library, starting on page 66), e.g.:

  ```
  12.01.56 JOB12345 FJM2100 FILE TRANSFERRED, TRANS_ID 1234567890
  ```

- Asynchronous messages output to a TSO terminal after a file transfer (see ENDMSG_TO_TSO in the PARM member of the FT parameter library, page 66), e.g.:

  ```
  12.01.56 JOB12345 SE 'FTR0005 OPENFT:Request 1234567890. File
  "DATASET.TEST" transferred, USER=...
  ```

- A list of all file-specific character sets used by openFT because of the specifications made in the member FNAMECTB of the FT parameter library (see page 89).

6.3 Internal openFT data sets

When certain FT administration commands are executed, openFT creates self-controlled internal data sets which are required for normal operation (logging file) or which contain diagnostic data (dump and trace files). These data sets must be deleted explicitly (dump files).

When using FTAC, openFT also automatically creates FTAC files in which the data generated and needed by FTAC are stored.

In total, the following internal openFT data sets exist for each openFT instance:

<openft qualifier>.<inst>.SYSRQF
: Request queue (DA data set on disk)

<openft qualifier>.<inst>.SYSPTF
: Partner list, corresponds to what used to be the network description file (DA data set on disk)

<openft qualifier>.<inst>.SYSOPF
: Operational parameters file (DA data set on disk)

<openft qualifier>.<inst>.SYSLOG.Lyymmdd.Lhhmmss etc.
: Components of the logging file:

  <openft qualifier>.<inst>.SYSLOG.Lyymmdd.Lhhmmss: PS data set

  <openft qualifier>.<inst>.SYSLOG.Lyymmdd.Lhhmmss.P00,
  <openft qualifier>.<inst>.SYSLOG.Lyymmdd.Lhhmmss.P00.D,
  <openft qualifier>.<inst>.SYSLOG.Lyymmdd.Lhhmmss.P00.I: Components of a VSAM cluster

  In this example a 7-character openFT qualifier and a 5-character instance name were used. If the openFT qualifier is any longer or shorter, the display of the components can differ from this pattern.
  yymmdd is the date (year, month, day) and hhmmss the time (hour, minute, second) when the logging file was created.

  Instead of the second level qualifier <inst>.SYSLOG used as a standard, a value specified by the administrator can be used (keyword LOGFILE_2ND_Q in the member PARM of the FT parameter library, see page 60). Note that depending on the length of the openFT qualifier and, possibly, of the "second level qualifier" the timestamps in the log file names are truncated or, in extreme cases, completely omitted.

  All FT and FTAC logging records are stored in that file. If the file is deleted or corrupted by individual components being deleted, the logging records are all lost.

If problems occur when the logging file is created or when it is being accessed, openFT cannot be started. (Such problems might arise from there being insufficient storage space or due to access protection for the file; refer to section “Protecting openFT administrative files” on page 39 for further information on admission protection.) The openFT job log file will contain the following message for example:

```
FTR0855 OPENFT: No space left on device for internal files
```

The FT system creates the logging file with the primary allocation, which you specified with the option LOGFILE_SIZE_RC (member PARM in the FT parameter library, see page 60); the value is halved for the size of the secondary allocation. The total size of the logging file depends on the number of logging records is contains. In your role as FT administrator, depending on the volume of requests, you should save the existing logging records from time to time and then delete them from the system using FTDELLOG (see description of the command FTDELLOG). This preserves contiguous documentation of the logging record over a longer period of time, while at the same time freeing storage space. Note that the allocated file size does not change. The space no longer occupied within the file is simply released again.

We recommend you use the following command from time to time to check to what extent the file contents has been split up:

```
LISTCAT ENT('<openft qualifier>.<inst>.SYSLOG.P00') ALL
```

If the file contents is split up to much, performance may deterioration. In that case the file should be reorganized, i.e. a new VSAM cluster with the same characteristics as the existing one should be created and the file contents should be copied using REPRO.

If no further logging records can be written into the logging file because it is full, the openFT instance system automatically deactivates itself (with the internal execution of the command FTSTOP). The openFT job log contains the system message IEC070I (meaning "An error occurred during EOV (end_of_volume) processing for a VSAM data set"). The FT administrator must then first make space available in the logging file by deleting logging records (command FTDELLOG), then the FT system can be reactivated (FJSTART) and FT operation can continue.

<openft qualifier>.<inst>.SYSFSA etc.

Components of the FTAC file (only when FTAC is used):

<openft qualifier>.<inst>.SYSFSA: PS data set

<openft qualifier>.<inst>.SYSFSA.P00,
<openft qualifier>.<inst>.SYSFSA.P00.DATA,
<openft qualifier>.<inst>.SYSFSA.P00.INDEX,
<openft qualifier>.<inst>.SYSFSA.P01,
<openft qualifier>.<inst>.SYSFSA.X01,
<openft qualifier>.<inst>.SYSFSA.X01.DATA,
<openft qualifier>.<inst>.SYSFSA.X01.INDEX: Components of a VSAM cluster

Instead of the second level qualifier <inst>.SYSFSA used as a standard, a value specified by the administrator can be used (keyword FILE_2ND_Q in the member FTACPAR of the FT parameter library, see page 91).

The file contains the FTAC environment, i.e. the admission sets, admission profiles, etc. If the file is deleted or corrupted by individual components being deleted, all admission profiles and admission sets are lost.
If problems occur when the FTAC file is created or when it is being accessed, openFT cannot be started. (Such problems might arise from there being insufficient storage space or due to access protection for the file; refer to section “Protecting openFT administrative files” on page 39 for further information on admission protection.)
The openFT job log file will contain the following message for example:

```
FTR0855 OPENFT: No space left on device for internal files
```

The FT system creates the FTAC file with the primary allocation which you specified in the parameter FILE_SIZE_KB (member FTACPAR of the FT parameter library, see page 91); the value is halved for the size of the secondary allocation.

We recommend you use the following command from time to time to check to what extent the file contents has been split up:

```
LISTCAT ENT('<openft qualifier>.<inst>.SYSFSA.P00') ALL
```

If the file contents is split up to much, performance may deterioration. In that case the file should be reorganized, i.e. a new VSAM cluster with the same characteristics as the existing one should be created and the file contents should be copied using REPRO.

If an FTAC command with which data are to be stored in the FTAC file fails (e.g. the command FTCREPRF) because the file is too full, the command is rejected and the following message is issued:

```
FTC0255 CMD TERMINATED. SYSTEM ERROR
```

The FT system automatically deactivates itself (with the internal execution of the command FTSTOP). The openFT job log contains the system message IEC070I (meaning "An error occurred during EOV (end_of_volume) processing for a VSAM data set").The FT system only needs to be restarted (FTSTART) in order to continue FT operation. However, before any further information can be stored in the FTAC file, the FTAC administrator must make space available in the FTAC file by deleting admission sets and admission profiles that are no longer required.
The FTAC administrator can find out which admission profiles exist by having them displayed with the command FTSHWPRF. Admission profiles are deleted with the command FTDELPRF.
There is no special command for deleting admission sets. The FTAC administrator deletes an admission set by setting the admission set for the relevant user ID back to the standard admission set (command FTMODADS, with MAX-LEVELS=*STD). This

can also be done for user IDs that have already been deleted. The FTAC administrator can find out which user ID have an admission set that differs from the standard admission set with the command FTSHWADS.

<openft qualifier>.<inst>.SYSFDF.Ddddmmmm
openFT dump file (PS data set on disk)
Dump information in this file is written automatically when a serious openFT error is encountered (e.g. protocol infringement, error situations where the messages FTR4024ff are issued, incorrect behavior of an openFT-specific exit routine).

'<openft qualifier>.<inst>.Smddhhmm.Sssccc.Iiii..FTTF" or
'<openft qualifier>.<inst>.Ymddhhmm.Sssccc.Pnnnnnnn.FTTF'
openFT trace file (FT trace file, PS dataset with 'Undefined' record format on disk). (see the FTMODOPT command)

Explanations

Depending on the length of the openFT qualifier , parts of the timestamp may be missing.

<openft qualifier>
OPENFT QUALIFIER that was defined using the FJGEN command (see page 113)

<inst> Name of the openFT- instance

ddd day of the year

mmmm minute of the day

mddhhmm timestamp in the format month-day-hour-minute (month: 1 = January, 2 = Februar, ... A = October, B = November, C = December)

ssccc continuation of the timestamp in the format seconds-milliseconds

iii index of the server process or 000 for the control process

nnnnnnn process ID

The request file, the partner list, the log file and the FTAC file are set up on the volume specified for this purpose in the PARM member of the FT parameter library (keywords NABVOLUME/NABUNIT). If no specification is made here, these files are set up on the same volume as the trace and dump files. You can define this volume, too, in the PARM member of the FT parameter library (keywords DMP_VOLUME/DMP_UNIT). If no specification is made here either, these files are set up on the volume specified for VOLUME/UNIT in the FJGEN command. The dump and trace files are set up on the volume specified for this purpose in the PARM member of the FT parameter library (keywords DMP_VOLUME/DMP_UNIT). If no specification is made here, the dump and trace files are set up on the volume specified for VOLUME/UNIT in the FJGEN command.

6.4 Temporary openFT data sets

In order to execute certain functions, openFT creates temporary data sets. These are normally deleted automatically after the function has been executed. If, due to an error, they are retained, they must be deleted explicitly. The temporary openFT data sets are as follows:

transuid.podsname.U
: Each time an entire PO or PDSE data set is transferred, a temporary PS data set is created in the send and receive system as a buffer for the file in "unloaded" format (IEBCOPY). These data sets are normally deleted after transfer.

  transuid: User ID specified in the TRANSFER-ADMISSION for the system involved.

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

  .U: This suffix identifies the temporary PS data set.

  These temporary PS data sets are set up on the volume specified for this purpose in the PARM member of the FT parameter library (keywords UNLOADVOL/UNLOADUNIT). If no specification is made here, the system defaults for newly created files apply.

  Each of these temporary PS data sets has approximately the same storage requirements as the corresponding PO data set.

<openft qualifier>.IEBSPILL.ddn1.ddn2
: Each time an entire PO or PDSE data set is transferred, small temporary PS data set is created in the send and receive system as a buffer for creating the directory in the event of a main memory bottleneck. These data sets are normally deleted after transfer.

  <openft qualifier>: OPENFT QUALIFIER that was defined using the FJGEN command (see page 113)

  ddn1, ddn2: DD name supplied by the z/OS system.

  These scratch files are set up on the default volume (system-specific).

FJCMD.TMP.OUT
: When some menu interface functions are executed, a temporary PS data set is created as a buffer for the command. This data set is normally deleted after execution of the function.

  These scratch files are set up on the default volume (system-specific).

<openft qualifier>.<inst>.S.PP.@num.id.ERR
<openft qualifier>.<inst>.S.PP.@num.id.OUT

Scratch files that are created during preprocessing and postprocessing are usually then deleted again.

<openft qualifier>:
OPENFT QUALIFIER that was defined using the FJGEN command (see page 113)

<inst>: Instance name of the currently set openFT instance

6.5 FT system messages

The structure of the FT system messages is as follows:

```
FTRnnnn OPENFT: message text
```

or

```
FTCnnnn message text
```

FTRnnnn

is the message code. The message code is 7 characters long.

message text

is the message text. The text appears in uppercase letters. The message text can contain what are known as inserts, e.g. (&00). These parts of the messages are supplied with the current value (e.g. transfer ID) when the message is output.

Additional explanatory information for the message is given under "Meaning"; "Response" tells you what action you should take. The texts are not displayed with the message.

Messages with the message code **FTRnnnn** (nnnn < 4000) and **FTCnnnn** are displayed both for the FT user and for the FT administrator.

Messages with the message code **FTR4nnn** are only displayed for the FT administrator. Messages with this message code are only described in this manual.

All message lists were generated with a view to your finding in them any error message that you might encounter. Consequently, the lists also contain a series of messages, that are only output under very specific circumstances (i.e. very rarely).

This also means that you cannot always expect the message from among those contained in the lists that would seem most appropriate to you.

Asynchronous messages for the FT administrator

Messages are normally a response to administration commands. There are, however, messages that are not generated by, or not only by, administration commands. These messages are output when FT administration mode is activated and/or to a console/a number of consoles and written to the openFT job log.

The meaning of these messages and the responses are explained on page 201.

Asynchronous messages for the user

In addition to the request confirmation or rejection, which is output to the user directly after the command has been entered, the FT system can also send asynchronous messages to the user indicating that file transfer has been terminated.

Further information is given in the manual "openFT (z/OS) - Command Interface".

Error codes and additional information

Additional error codes and supplementary partner-specific information can be output for some user messages. These provide additional information for troubleshooting.

As a rule, this supplementary information is made up of a return code from the operating system together with a text supplied by the operating system issued in the language set in the operating system. The supplementary information can also comprise an English language text generated by openFT itself.

It is appended to the end of the message and is restricted to a length of 64 characters. Longer texts are truncated.

6.5.1 FTR4nnn messages

FTR4004 OPENFT: SMF NOT ACTIVE.

Meaning
When an attempt was made to write an accounting record to the SMF file, SMF was found to be inactive. No more accounting records are written for any subsequent transfer requests.

FTR4005 OPENFT: SMF ERROR.

Meaning
An error (possibly a temporary one) occurred when writing an accounting record.

FTR4006 OPENFT: SMF RECORDING STOPPED.

Meaning
No more accounting records are written for any subsequent transfer requests because errors occurred during 10 consecutive attempts to write a record to the SMF file or SMF was found to be inactive or no longer active.

FTR4010 OPENFT: INQUIRE FOR COMMON BUFFER SPACE FAULTY

Meaning
Not enough memory could be reserved for buffering a command entered at an operator console (asynchronous message issued to the FT administrator).

Response
Ask the system administrator.

FTR4026 OPENFT: CONSOLE–TASK EVENTING ERROR

Meaning
The console task has received an unexpected event. (This error message only appears in the openFT job log. When this error occurs, a dump is generated and written to the file SYSFDF.Ddddmmmm. openFT continues to execute, however.)

Response
Ask the system administrator.

FTR4041 OPENFT: ERROR OCCURRED WHILE READING TNSTCPIP.

Meaning
An error occurred while the TNSTCPIP member was being read in from the FT parameter library.

Response
If openFT is to be connected to remote systems via TCP/IP: Make sure that the FT parameter library and TNSTCPIP member both exist and that openFT can access them.

FTR4042 OPENFT: SYNTAX ERROR IN TNSTCPIP ENTRY.

Meaning
A syntax error was discovered in one of the data records in the TNSTCPIP member of the FT parameter library. The defective data record is ignored and reading of the TNSTCPIP member is continued. The field in which the syntax error was first identified is generally indicated by means of one of the messages below (FTR4043 to FTR4046).

Response
Correct the syntax error (see additional message) and enter the FTUPDPAR command.

FTR4043 OPENFT: ILLEGAL TNS-NAME (&00).

Meaning
Illegal syntax has been used for the TNS name (&00) (additional message to message FTR4042).

Response
Correct the TNS name (name of address entry). A valid TNS name (name of the address entry) must be unique within the local system and consist of a maximum of 8 alphanumeric characters, the first of which must be a letter or one of the special characters $, @ or #.

FTR4044 OPENFT: ILLEGAL INTERNET ADDRESS (&00).

Meaning
Illegal syntax has been used for the Internet address (&00) (additional message to message FTR4042).

Response
Correct the Internet address. A valid Internet address has the form xxx.xxx.xxx.xxx, where xxx is an integer (in decimal representation) in the range 0 to 255.

FTR4045 OPENFT: ILLEGAL PORT NUMBER (&00).

Meaning
Illegal syntax has been used for the port number (&00) (additional message to message FTR4042).

Response
Correct the port number. A valid port number consists of an integer in the range 1 to 32767.

FTR4046 OPENFT: ILLEGAL TSEL (&00).

Meaning
Illegal syntax has been used for the T-selector (&00) (additional message to message FTR4042).

Response
Correct the T-selector. A valid T-selector can consist of up to 32 characters.

FTR4048 OPENFT: TNSTCPIP RECORD LIMIT EXCEEDED.

Meaning
The TNSTCPIP member contains more than 10000 records. All records after record number 10000 are ignored.

Response
This message can be ignored if you are sure that the specifications for all partner systems that are to be accessed via TCP/IP are contained in the first 10000 records of the member.

FTR4053 OPENFT: CONNECTION TO TCP/IP SEVERED. REASON = (&00)

Meaning
The openFT connection to TCP/IP activated previously has been aborted. File transfers via TCP/IP are no longer possible.
Either a detailed reason is given as reason code in this message or the original reason code of the software product used for the link to TCP/IP (TCP/IP (for MVS) from IBM or compatible product).

Response
First, the cause of the unwanted termination of the software product used for the TCP/IP link must be removed.
Since openFT itself does not try to restore the connection to TCP/IP, the FT system must then be deactivated (FTSTOP command) and reactivated again (FTSTART command).

FTR4054 OPENFT: MYPORT-NUMBER (&00) NOT AVAILABLE

Meaning
A value for the openFT passive port that is already used by another TCP application was specified in the PORT parameter of the FJGEN command.

Response
Either terminate the TCP application that is occupying this port number or deactivate openFT (FTSTOP), correct the PORT specification in the FJGEN command and reactivate openFT (FTSTART).

FTR4055 OPENFT: TCP/IP-TRANSPORT NOT ENABLED. REASON=(&00)

Meaning
openFT was unable to log on to the TCP/IP transport system. The TCP/IP address space cannot be accessed.

Response
Check the generation for connecting openFT to TCP/IP. You may have to specify or check the name of the TCP/IP address space (TCP_USERID in the PARM member of the parameter library). If you are unable to find the reason for the error, report the error and the reason (&00) contained in the message to your Service Center.

FTR4056 OPENFT: HOST NAME (&OO) UNKNOWN

FTR4057 OPENFT: LOCAL IP-ADDRESS (&OO) NOT SUPPORTED

FTR4120 OPENFT: INITIATED

Meaning
The openFT instance has been loaded in response to the FJINIT command.

FTR4121 OPENFT: TERMINATED

Meaning
The openFT instance has been unloaded in response to the FTTERM command, or abnormally terminated due to a serious error.

FTR4125 OPENFT: PARAMETERS TNSTCPIP, FTADM and FTACADM UPDATED

Meaning
The parameters have been successfully updated using the FTUPDPAR command.

FTR4131 OPENFT: TERMINATION INITIATED BY USER

Meaning
A user has entered the FTTERM command under TSO.

FTR4140 OPENFT: TERMINATED. MAX NO. OF INSTANCES EXCEEDED.

Meaning
An attempt was made to load a 17th openFT instance.

Response
Terminate another instance with FTTERM or Cancel.

FTR4141 OPENFT: TERMINATED. INSTANCE NAME IN USE.

Meaning
An attempt was made to load openFT with an instance name that is already in use.

Response
Use a different instance name.

FTR4144 OPENFT: CMD REJECTED. USER NOT AUTHORIZED

Response
An NCOPY command was entered by a job for which there is neither a user ID ("user-id.") or a "dsname prefix" or a command was entered that the caller does not have permission to issue.

FTR4145 OPENFT: CMD REJECTED. SESSION IDENTIFICATION FAILED

FTR4150 OPENFT: CMD REJECTED. SYNTAX ERROR

Meaning
This message is output if the command is entered with incorrect syntax (e.g. command name written wrongly).

FTR4180 OPENFT: CMD REJECTED. INTERNAL SYSTEM CALL FAILED

FTR4192 OPENFT: (&00) NOT KNOWN TO TRANSPORT SYSTEM

Meaning

a) An application, LU or LOGMODE name (&00) was not found in the generation of the transport system (VTAM). If this message is issued for an NCOPY/NCANCEL/NSTATUS command (&00 = FJNNDMS0, FJNNDMS1,...), the specified LU has not been generated or all generated LUs of this type are currently reserved.

b) The attempt to determine the Internet address of a remote computer from its host name (&00) via the z/OS Name Services, i.e. either via the "Domain Name System" (DNS) or the file TCPIP.HOSTS.LOCAL ("Flat Name Space"), has failed.

FTR4193 OPENFT: (&00) NOT AVAILABLE

Meaning
An application or openFT is currently not available. If this message is issued during processing of an NCOPY/NCANCEL/NSTATUS command and (&00)=LU, all LUs of the type FJNNDMSx are generated and currently reserved. If this message is issued during processing of an NCOPY/NCANCEL/NSTATUS command and (&00)=FJNNDMSx, a VTAM open error has occurred for this LU.

FTR4196 OPENFT: DIALOG HANDLER VERSION INCOMPATIBLE

FTR4197 OPENFT: (&00) TERMINATED BY TIMEOUT

FTR4199 OPENFT: SYSTEM ERROR. ERROR CODE (&00)

Meaning
An operating system function called by openFT has reported an error. The error code (&00) has two halves:

left half	operating system function used.
right half	return code of the operating system function used.

FTR4200 OPENFT: FTADM VERSION (&00) INITIATED.

FTR4201 OPFT SUBSYSTEM NOT INSTALLED, BUT SPECIFIED

Meaning
The parameter OPENFT_SVC is set in the PARM member of the FT parameter library but openFT is not installed as a subsystem.

Response
Install openFT as a subsystem or comment out OPENFT_SVC in the PARM member.

6.5.2 FTR messages

FTR0000 OPENFT: Request (&00) accepted.

Meaning
The command has been stored in the local system's request queue. File transfer will begin once all the resources have been assigned in both the local and remote system.

(&00): transfer ID assigned by the local FT system. You need the transfer ID in case you wish to cancel (NCANCEL) the FT request later.

FTR0005 OPENFT: Request (&00). File '(&01)' transferred.

Meaning
The file transfer request (&00) has been completed successfully. Follow-up processing for both the local and remote system, if requested, has been initiated (provided no error occurred). Local Errors are indicated by a message.

FTR0006 OPENFT: Request (&00). Directory '(&01)' transferred.

Meaning
The directory transfer request (&00) has been completed successfully. Follow-up processing for both the local and remote system, if requested, has been initiated (provided no error occurred). Local Errors are indicated by a message.

FTR0020 OPENFT:'(&00)' not found.

Meaning
The command has not been executed because the send file is not cataloged or not on a volume of the local system. The command has not been executed because either the send file is not/is no longer, or the receive file is no longer in the catalog or on a volume of the relevant system.

Response
Correct the file name, read in file from tape or restore send file. Repeat the command.

FTR0035 OPENFT: File locked to prevent multiple access.

Meaning
The command has not been executed because either the send file or the receive file is already locked by another process against simultaneous updating.

Response
Repeat the command later or unlock the file. After a system crash you may need to verify files that are not closed correctly.

FTR0041 OPENFT: Request queue full.

Meaning
The command has not been executed because the maximum number of permissible transfer requests has been reached.

Response
Notify the FT administrator. Repeat the command later.

FTR0108 OPENFT: Request (&00). Remote system not accessible.

Meaning
The command could not be accepted because the partner system is currently not available.

Response
Repeat the command later. If the error persists, contact the system or network administrator.

FTR0236 OPENFT: Current instance (&00) no longer found

Meaning
The command was rejected. The instance (&00) could not be found.

FTR0301 OPENFT: Partner '(&00)' entered state NOCON.

Meaning
The partner system (&00) has switched to the state NOCON. This state means that the partner is no longer accessible.

Response
If necessary, check whether the connection to the partner system has been interrupted.

FTR0302 OPENFT: Partner '(&00)' entered state ACTIVE.

Meaning
The partner system (&00) has switched to the state ACTIVE.

Response
For information only.

FTR0303 OPENFT: Partner '(&00)' entered state LUNK.

Meaning
The partner system (&00) has switched to the state LUNK. This state means that the local FT system is not known in the remote FT system.

Response
Ask the remote system's FT administrator to enter the local system in the remote system's network description file/partner list.

FTR0304 OPENFT: Partner '(&00)' entered state RUNK.

Meaning
The partner system (&00) has switched to the state RUNK. The state RUNK means that the remote system is not known in the local transport system.

Response
Make the remote system known on the local system.

FTR0305 OPENFT: Partner '(&00)' entered state INACT.

Meaning
The partner system (&00) has switched to the state INACT. The state INACT means that the FT administrator has locked outbound requests for this partner system.

Response
Remove the lock if necessary.

FTR0306 OPENFT: Partner '(&00)' entered state AINACT.

Meaning
The partner system has switched to the state AINACT. The state AINACT means that the partner system has been automatically deactivated because a certain number of consecutive connection attempts have failed.

Response
Check whether partner system should be accessible and reactivate the partner system.

FTR0307 OPENFT: Partner '(&00)' may be unreachable.

Meaning
A number of consecutive attempts to connect to the partner system (&00) have failed. Further attempts will be made.

Response
For information only.

FTR0308 OPENFT: Partner '(&00)' does not allow more inbound requests.

FTR0309 OPENFT: Partner '(&00)' added.

Meaning
The specified remote system has been entered in the partner list.

FTR0310 OPENFT: Partner '(&00)' removed.

Meaning
The specified remote system has been removed from the partner list.

FTR0311 OPENFT: Partner '(&00)' entered state LAUTH.

Meaning
The partner system (&00) has switched to the state LAUTH. The state LAUTH means that the local system could not authenticate itself at the remote system.

Response
Send the current key file to the administrator of the remote system.

FTR0312 OPENFT: Partner '(&00)' entered state RAUTH.

Meaning
The partner system (&00) has switched to the state RAUTH. The state RAUTH means that the remote system could not authenticate itself at the local system. This may either be due to an out-of-date key in the key file or to may indicate an access attempt by an unauthorized system.

Response
Contact the administrator of the remote system.

FTR0313 OPENFT: Partner '(&00)' entered state DIERR.

Meaning
The partner system (&00) has switched to the state DIERR. File integrity errors have been detected on the transmission path. This may also indicated deliberate manipulation of the transmission data.

FTR0314 OPENFT: Partner '(&00)' entered state NOKEY.

Meaning
The partner system (&00) has switched to the state NOKEY. The state NOKEY means that the partner will not accept a connection without encryption or that no key is present.

Response
Generate a new key pair.

FTR0315 OPENFT: Partner '(&00)' entered state IDREJ.

Meaning
The partner system (&00) has switched to the state IDREJ. The local identification was not accepted by the local identification or by an intermediate entity.
Possible causes:
– both the local identification and the migrated ID %.<processor>.<entity> are entered in the remote system's request file.
– the identification has been rejected by an intermediate entity for security reasons

Response
Ask for your entity's partner entry to be checked.

FTR0320 OPENFT: Abnormal termination initiated.

Meaning
Abnormal termination of FT has been initiated due to an internal error.

Response
Check the cause of the abnormal termination and restart FT.

FTR0330 OPENFT: Request queue 85 percent full.

Meaning
Approximately 85% of the spaces for request storage in the request file are occupied. Issuing a number of additional requests could completely fill the request queue with the result that FT will reject new requests.

Response
If necessary, increase the size of the request queue.

FTR0331 OPENFT: At least 20 percent of request queue unoccupied.

Meaning
At least 20% of the FT request queue is available. This message is only output if a previous FTR0330 message has warned of a possible queue overflow. The threat of a bottleneck has receded.

FTR0340 OPENFT: Transfer '(&00)' successfully completed.

Meaning
The request designated in greater detail by the insert (&00) has been terminated successfully.
(&00): *LOC/*REM;SID;PARTNER;USERID;FILE
Since the length of the insert is limited to a maximum of 180 characters, the file name may be truncated if necessary. This is indicated by the character '*' at the end of the file name.

Response
For information only.

FTR0341 OPENFT: Transfer '(&00)' terminated with error.

Meaning
The request designated in greater detail by the insert (&00) terminated with an error
(&00): MSGNR;*LOC/*REM;SID;PARTNER;USERID;FILE
Since the length of the insert is limited to a maximum of 180 characters, the file name may be truncated if necessary. This is indicated by the character '*' at the end of the file name.

Response
For information only.

FTR0360 OPENFT: openFT control process started

Response
For information only.

FTR0361 OPENFT: openFT control process terminated

Response
For information only.

FTR0500 OPENFT: openFT (&00) started. Protocols: (&01).

Meaning
The openFT file transfer system openFT has been activated for the protocols (&01).

FTR0501 OPENFT: openFT terminated.

Meaning
The file transfer system openFT has been terminated by means of an administration command.

FTR0502 OPENFT: No log records available for the selection criteria.

Meaning
No logging records meet the selected criteria.

Response
Change the selection criteria.

FTR0503 OPENFT: No partner available for the selection criteria.

Meaning
There are no partners that meet the specified selection criteria.

Response
Change the selection criteria.

FTR0504 OPENFT: No requests available for the selection criteria.

Meaning
There are no requests that meet the specified selection criteria.

Response
Change the selection criteria.

FTR0505 OPENFT: Requests carried out; (&00) files were transferred

Meaning
The file transfer requests have been successfully completed. A total of (&00) files have been transferred. If you have specified commands for follow-up processing, follow-up processing is carried out for every file.

FTR0509 Protocol: (&00) not installed or not licensed

Meaning
Configuration and activation of protocol don't match.

Response
Check configuration and activation of protocol.

FTR0510 OPENFT: Requests carried out; (&00) directories were transferred

Meaning
The directory transfer requests have been successfully completed. A total of (&00) directories have been transferred. If you have specified commands for follow-up processing, follow-up processing is carried out for every file.

FTR0511 OPENFT: Requests accepted; (&00) file transfers initiated

Meaning
The file transfer requests have been successfully accepted.
A total of (&00) file transfers have been initiated. If you have specified commands for follow-up processing, follow-up processing is carried out for every file.

FTR0560 OPENFT: Cancel all specified requests? Reply (y=yes; n=no)

Meaning
A CANCEL-TRANSFER command applies to more than one file transfer.
Y: All the transfer requests affected are deleted.
N: The entire deletion request is withdrawn.

FTR0561 OPENFT: Do you really want to delete this key pair set? Reply (y=yes; n=no)

Meaning
Deletion of a key pair set prevents connection to all partners working with these keys. If the last key pair set is deleted, there is no encryption.

FTR0562 OPENFT: (&00):

FTR0600 OPENFT: Shutdown processing delayed. FT tasks pending.

Meaning
openFT could not be terminated.

Response
Check if there are console messages that need to be answered for FT tasks connected to the FT subsystem.

FTR0604 OPENFT: Request (&00). Follow-up processing not started.

Meaning
The follow-up processing of a transfer request was not started because the local processing admission may be incorrect.

Response
Correct the local processing admission and repeat the command.

FTR0605 OPENFT: Tracefile changed

Meaning
There has been a switch to a new trace file.

FTR0606 OPENFT: Trace terminated.

Meaning
The trace status has been switched off.

FTR0607 OPENFT: Trace started: (&00).

Meaning
The trace status for the protocols specified in (&00) has been switched on.

FTR0700 Parameter '(&00)' and '(&01)' must not be specified at the same time

Meaning
The selected parameters could not be specified simultaneously.

Response
Omit one of the two parameters and repeat the command.

FTR0701 OPENFT: Input error

FTR0702 OPENFT: Parameter value '(&00)' too long

Meaning
The specified parameter value (&00) is too long; see the command syntax.

Response
Reduce the length of the parameter value (&00) and repeat the command.

FTR0703 OPENFT: Mandatory parameter missing

Meaning
A mandatory parameter is missing; see the command syntax.

Response
Correct the command and try again.

FTR0704 OPENFT: Mandatory parameter '(&00)' missing

Meaning
The mandatory parameter (&00) was not specified.

Response
Correct the command and try again.

FTR0705 OPENFT: Parameter '(&00)' specified more than once

Meaning
The parameter (&00) was specified more than once.

Response
Correct the command and try again.

FTR0706 OPENFT: Parameter '(&00)' can only be specified together with '(&01)'

Meaning
The parameter (&00) can only be specified together with (&01).

Response
Add the parameter (&01) to the command and repeat the command.

FTR0707 OPENFT: Invalid parameter '(&00)'

Meaning
An invalid parameter (&00) was specified; see the command syntax.

Response
Correct the command and try again.

FTR0708 OPENFT: Value of parameter '(&00)' not within valid range

Meaning
The parameter value (&00) is not within the specified value range; see the command syntax.

Response
Correct the parameter value (&00) and repeat the command.

FTR0709 OPENFT: Too many positional parameters

Meaning
The maximum number of positional parameters was exceeded.

Response
Correct the command and try again.

FTR0710 OPENFT: Invalid parameter value '(&00)'

Meaning
The assigned parameter value (&00) is incorrect; see the command syntax.

Response
Correct the parameter value (&00) and repeat the command.

FTR0711 Parameter '(&00)' or '(&01)' must be specified

Meaning
At least one of the parameters must be specified.

FTR0750 OPENFT: Command not found

FTR0751 OPENFT: Command name ambiguous with regard to '(&00)'

FTR0752 OPENFT: Closing parenthesis missing for operand '(&00)'

FTR0753 OPENFT: Invalid delimiter '(&00)' after operand '(&00)'

FTR0755 OPENFT: List value of operand '(&00)' is not consistent with data type '(&00)'

FTR0756 OPENFT: Operand value introducing the structure is mandatory for '(&00)'

FTR0757 OPENFT: Value of operand '(&00)' is not consistent with data type '(&00)'

FTR0758 OPENFT: Keyword value of operand '(&00)' is ambiguous with regard to '(&00)'

FTR0759 OPENFT: Too many closing parentheses

FTR0760 OPENFT: The mandatory operand '(&00)' is missing

FTR0762 OPENFT: Operand name '(&00)' ambiguous with regard to '(&00)'

FTR0763 OPENFT: Operand '(&00)' is not known

FTR0764 OPENFT: Operand '(&00)' specified more than once

FTR0765 OPENFT: Too many list elements for operand '(&00)'

FTR0766 OPENFT: Too many positional operands

FTR0767 OPENFT: Too many positional operands for '(&00)'

Meaning
(applies to FTR0750 through FTR0767)
An operand value that introduces a structure can only be omitted if there is only one possible structure specification for the corresponding operand or if this structure specification is the default value for the operand.
The following command, for example, will be rejected with this message:

```
FTMODPRF MYPROF01,PARTNER=((REMSYS1,REMSYS2))
```

Reason: It is not clear which of the following specifications is meant:

```
FTMODPRF MYPROF01,PARTNER=*ADD((REMSYS1,REMSYS2))
```

or

```
FTMODPRF MYPROF01,PARTNER=*REM((REMSYS1,REMSYS2))
```

Response
Repeat the command using the correct syntax.

FTR0780 OPENFT: Internal error: operand buffer overflow

FTR0781 OPENFT: Internal error: structure nesting too deep

FTR0790 OPENFT: Available commands: '(&00)'

FTR0791 OPENFT: Available list-values: '(&00)'

FTR0792 OPENFT: Available operands: '(&00)'

FTR0793 OPENFT: Available values: '(&00)'

FTR0801 OPENFT: Request (&00). Internal error

Meaning
NDMS, FJAM or operating system error that is neither a DMS error nor a transport system error, possibly the transfer ID.
The FT system continues to run after the message has been issued.

FTR0802 OPENFT: Request (&00). Warning: Monitor file contents inconsistent

Meaning
At the end of the file transfer request, the contents of the job variable monitoring the request were found to be inconsistent.
Possible reason: During the transfer, the job variable was accessed externally in a mode other than read mode.
The result of the transfer is not affected and is given in the result list or asynchronous end message.

FTR0803 OPENFT: Request (&00). Follow-up processing could not be started.

Meaning
The command was not executed because the specifications in one of the PROCESSING-ADMISSION operands are incorrect.

Response
Define the required PROCESSING ADMISSION or correct it. Repeat the command if necessary.

FTR0804 OPENFT: Request (&00). Request data inconsistent.

FTR0851 OPENFT: Internal error.

FTR0852 OPENFT: Internal error. Current instance '(&00)' incompatible.

Meaning
The system data was not created with the version of the openFT file transfer system currently in use.

Response
Update the instance to the current openFT version using the appropriate command (FJGEN).

FTR0854 OPENFT: Writing of log records no more possible. Process terminated.

Meaning
There is not enough space on the disk/partition on which the logging files are stored.

Response
Increase the disk space (or have it increased).

FTR0855 OPENFT: No space left on device for internal files.

Meaning
There is not enough space on the disk/partition on which the internal files are stored.

Response
Increase the disk space (or have it increased).

FTR0856 OPENFT: Error during ops generation.

FTR0857 OPENFT: Error in key file (&00)

FTR0858 OPENFT: Internal error. Set / release file-locks not possible

Meaning
A problem occurred when setting/resetting the file locks for all open requests in FT-REQUEST-FILE.

Response
Check whether the request file SYSRQF is accessible on the config user ID of the current instance.

FTR0862 OPENFT: Protocol stack (&00) not installed

Meaning
The required transfer protocol is not installed.

Response
Install the transfer protocol.

FTR0863 OPENFT: FTAC subsystem not available

Meaning
Install openFT-AC.

FTR0864 OPENFT: Server (&00) accepts no requests

Meaning
The control process cannot access the server to assign new requests.

Response
Check if the server is blocked e.g. by a system dump or the output of a message and correct the error situation.

FTR0865 OPENFT: Not all files transferred successfully

Meaning
At least one source file could not be transferred to the local system. The previous messages indicate the files concerned:
Transfer of file '<file>' failed. Reason: '<rc>'

FTR0999 OPENFT: openFT panic (&00). Abnormal termination

FTR1020 OPENFT: openFT already started.

Meaning
openFT can only be started once in each instance.

Response
Terminate openFT if necessary.

FTR1021 OPENFT: Request must be canceled without FORCE option first

Meaning
Before the FORCE option is used, the command must be called without the FORCE option.

Response
Issue the command without the FORCE option first.

FTR1029 OPENFT: Maximum number of key pairs exceeded.

Meaning
The maximum number of key pair sets has been reached.

Response
Before new key pair set can be created, an older key pair set must be deleted.

FTR1030 OPENFT: Warning: last key pair deleted.

Meaning
The last key pair set has been deleted. Without a key pair set, encrypted transfer, authentication and data integrity checking are not possible.

Response
Create a new key pair set.

FTR1031 OPENFT: No key pair available.

Meaning
All transfers are carried out without encryption.

Response
Create a new key pair set, if necessary.

FTR1032 OPENFT: Last key pair must not be deleted

FTR1033 OPENFT: The public key files could not be updated.

Meaning
The contents of the SYSPKF file could not be fully updated.
Possible reasons:
– The SYSPKF file is locked.
– There is not enough disk space to allow the file to be created.

Response
Take the appropriate action depending on the cause of the error:
– Unlock the file.
– Allocate disk space or have your system administrator do it.

Update the key with UPDATE-FT-PUBLIC-KEY.

FTR1034 OPENFT: Command only permissible for FT or FTAC administrator

Meaning
Only the FT or FTAC administrator is permitted to use the command.

Response
Have the command executed by the FT or FTAC administrator.

FTR1035 OPENFT: Command only permissible for FT administrator.

Meaning
Only the FT administrator is permitted to use the command.

Response
Have the command executed by the FT administrator.

FTR1036 OPENFT: User not authorized for other user IDs.

Meaning
The user is not authorized to use a different user ID in the command.

Response
Specify your own ID, or have the command executed by the FT or FTAC administrator.

FTR1037 OPENFT: Key reference unknown.

Meaning
The specified key reference is unknown.

Response
Repeat the command with an existing key reference.

FTR1038 OPENFT: Request '(&00)' is in the termination phase and can no longer be canceled

FTR1039 OPENFT: openFT not active.

Meaning
openFT is not started.

Response
Start openFT, if necessary.

FTR1040 OPENFT: Config user ID unknown or not enough space

Meaning
The CONFIG USERID of the current instance (SYSFJAM) is unknown or the disk space allocated is insufficient to allow creation of the FT-REQUEST-FILE, the file for storing trace data, or the key files.

Response
Either create the CONFIG-USERID or increase its disk space allocation or have your system administrator do it.

FTR1041 OPENFT: Specified file is not a valid trace file

FTR1042 OPENFT: openFT could not be started

FTR1043 OPENFT: Partner with same attribute '(&00)' already exists in partner list.

Meaning
There is already a partner entry with the same attribute '(&00)' in the partner list.

Response
The attribute '(&00)' in partner entries must be unique. Correct the command accordingly and try again.

FTR1044 OPENFT: Maximum number of partners exceeded.

Meaning
The partner list already contains the maximum permissible number of partner entries.

Response
Delete partners that are no longer required.

FTR1045 OPENFT: No partner found in partner list.

Meaning
A partner for the specified selection could not be found in the partner list.

Response
Check if the specified partner name or address was correct. If necessary, repeat the command using the correct name or address.

FTR1046 OPENFT: Modification of partner protocol type not possible

Meaning
The protocol type of the partner entry cannot be changed subsequently.

Response
Delete the partner from the partner list, if necessary, and enter it again with a new protocol type.

FTR1047 OPENFT: Request (&00) not found.

Meaning
The request with the transfer ID (&00) could not be found.

Response
Specify the existing transfer ID and repeat the command.

FTR1048 OPENFT: Active requests could not yet be deleted

Meaning
Active requests for the specified partner were cancelled. After the negotiation of termination with the partner the requests will be automatically deleted.

FTR1049 OPENFT: CCS name (&00) unknown

FTR1057 OPENFT: Inbound requests cannot be modified

FTR1059 OPENFT: Monitoring is not active

Meaning
The command is only supported if monitoring is activated.

Response
Activate monitoring in the operating parameters.

FTR1060 OPENFT: File could not be created '(&00)'

FTR1061 OPENFT: Higher-level directory not found

FTR1062 OPENFT: File already exists

FTR1063 OPENFT: Resulting file name too long

FTR1064 OPENFT: File locked to prevent multiple access

FTR1065 OPENFT: File not found

FTR1066 OPENFT: Not enough space for file

FTR1067 OPENFT: Syntax error in resulting file name

FTR1068 OPENFT: Access to file denied (&00)

FTR1069 OPENFT: Error accessing file (&00)

FTR1073 OPENFT: Command aborted

Meaning
The user has interrupted the command.

FTR1076 OPENFT: selected key file not found

FTR1077 OPENFT: Change of transport access system not possible. Reason: (&00)'

FTR1078 OPENFT: Too short time interval since last logging file switch

Meaning
At this moment the logging file cannot be switched, as the time dependant part of the logging file name does not differ from this name part in the actual logging file name.

Response
If necessary, repeat the command after an appropriate waiting time.

FTR1082 OPENFT: User data encryption not supported

Meaning
User data encryption is supported only when openFT-CR is installed.

Response
Install openFT-CR

FTR1083 OPENFT: Structure of key file not supported

Meaning
The key cannot be imported because of the not supported key file structure.

FTR1084 OPENFT: Invalid password

FTR1085 OPENFT: Password missing

FTR1086 OPENFT: Duplicate key pair

Meaning
No import of duplicate keys allowed.

FTR1087 OPENFT: Key expired

Meaning
The expiration date lies in the past.

FTR1092 OPENFT: Warning: The RSA key length was aligned to the minimum key length

Meaning
The RSA key length was less than the configured minimum key length and had to be enlarged.

FTR2014 OPENFT: No file attribute changes requested.

Meaning
No further file attributes besides the file name were specified.

Response
Enter the desired file attributes in addition to the file name.

FTR2015 OPENFT: openFT is not authorized to execute requests for this user

FTR2016 OPENFT: Directory (&00) is not empty

FTR2017 OPENFT: File attributes do not match request parameters (&00)

Meaning
The specified attribute combination is not permissible.

Response
Specify a permissible combination.

FTR2018 OPENFT: Attributes could not be modified (&00).

Meaning
The properties of the file could not be changed as specified in the command.
The following reasons are possible:
For the remote file:
– No access rights to the file.
– The required combination of access rights is not supported by the remote system.
– If the remote system is a BS2000: the file is protected by ACL.

For the local file:
– No access rights to the file.
– The requested transfer attributes are not compatible with the properties of the file (see manual).

FTR2019 OPENFT: (&00)' could not be created (&01).

Meaning
The command was not executed because the file owner and user requesting the creation of a receive file are not the same.

Response
Match the user ID in the receiving system's TRANSFER-ADMISSION to the ID of the receive file's owner. Repeat the command.

FTR2021 OPENFT: CCS name unknown.

Meaning
The request could not be completed because the CCS name specified for the local file does not correspond to any of the supported code tables.

FTR2022 OPENFT: Higher-level directory not found

Meaning
In the case of a receive request, the local file could not be created because the specified path does not exist.

Response
Create or correct the path for the receive file and repeat the command.

FTR2023 OPENFT: (&00)' already exists.

Meaning
The command was not executed because an existing receive file cannot be created again with WRITE-MODE=NEW. WRITE-MODE=NEW may also have been set due to a restriction in the access authorization used.

Response
Either delete the receive file and repeat the command, or repeat the command specifying WRITE-MODE=REPLACE-FILE or using different access authorization.

FTR2024 OPENFT: Transfer of file generation groups not supported.

Meaning
The command was not executed because the FT system only transfers single file generations.

Response
Repeat the command using the name of a single file generation.

FTR2025 OPENFT: Error accessing '(&00)'(&02).

Meaning
(&02): Further details, possibly DMS error
The FT system continues to run after the message has been issued.

Response
Take the appropriate action in accordance with the error code.

FTR2026 OPENFT: Resulting file name '(&00)' too long (&01).

Meaning
The relative file name was specified in the transfer request. The absolute file name completed by openFT is longer than permitted.

Response
Shorten the file name or path and repeat the command.

FTR2027 OPENFT: No file or directory name specified.

Meaning
The command was not executed because the file name was neither specified explicitly nor by the 'TRANSFER-ADMISSION' used.

Response
Repeat the command, specifying the file ID explicitly or a TRANSFER-ADMISSION that defines the file ID.

FTR2028 OPENFT: Invalid management password.

FTR2029 OPENFT: (&00)' not available (&01).

Meaning
The command was not executed because the volume for either the send file or the receive file is not mounted, unknown or reserved, the file extends over more than one private disk, or an attempt has been made to transfer a file migrated by HSM without specifying the local transfer admission (TRANSFER-ADMISSION operand).

Response
Inform the operator if necessary or carry out an HSM recall for the file or specify the local transfer admission. Repeat the command.

FTR2030 OPENFT: Home directory not found (&00)

FTR2031 OPENFT: Renaming not possible (&00)

FTR2032 OPENFT: Not enough space for (&00).

Meaning
The command was not (fully) executed because the permissible storage space on the receive system is used up for the user ID specified in TRANSFER-ADMISSION.
The receive file can not be created/extended after the problem occurs.

Response
Take the appropriate action depending on the cause of the error:
– delete all files no longer required on the receive system, or
– ask the system administrator to allocate more storage space, or
– increase the receive file's primary/secondary allocation.

If WRITE-MODE=EXTEND-FILE is specified, restore the receive file.
Repeat the command.

FTR2033 OPENFT: File owner unknown.

Meaning
The command was not executed because the owner of either the send file or the receive file was not defined in the local system or because the file owner and the user requesting the creation of a receive file are not the same.

Response
Define the file owner, correct TRANSFER-ADMISSION or FILE-NAME.
Repeat the command.

FTR2034 OPENFT: Invalid file password.

Meaning
The command was not executed because the password for the send file or the receive file is missing or incorrect.

Response
Correct the password in the file description or the command.
Repeat the command.

FTR2036 OPENFT: Retention period of file not yet expired.

Meaning
The command was not executed because the retention period protecting the receive file against overwriting has not yet expired (RETENTION PERIOD).

Response
Correct the transfer direction, retention period or file name. Repeat the command.

FTR2037 OPENFT: '(&00)' is read only.

FTR2038 OPENFT: File structure not supported (&00).

FTR2039 OPENFT: Syntax error in resulting file name '(&00)' (&01).

Meaning
The local file cannot be accessed because, for example, the absolute file name is too long.

Response
Shorten the path or file name. Repeat the command.

FTR2040 OPENFT: Transparent file transfer not supported.

Meaning
The request could not be carried out because the partner system does not support the receipt of files in a transparent format.

FTR2042 OPENFT: Extension of file not possible for transparent transfer.

Meaning
The command could not be executed because it is not possible to add to a file in a transparent transfer.

Response
Start transfer without EXTEND.

FTR2043 OPENFT: Access to '(&00)' denied (&01).

Meaning
The command was not executed because either the send file or the receive file only permits certain access modes (e.g. read only).

Response
Correct the file name or file protection attributes. Repeat the command.

FTR2044 OPENFT: Follow-up processing exceeds length limit.

Meaning
Prefix + suffix (from prof) + local follow-up processing together are too long.

Response
Correct the file name or file protection attributes. Repeat the command.

FTR2045 OPENFT: Processing admission invalid.

Meaning
The command was not executed because the specifications in one of the PROCESSING-ADMISSION operands were incorrect.

Response
Define the required PROCESSING ADMISSION or correct it.
Repeat the command if necessary.

FTR2046 OPENFT: Local transfer admission invalid.

Meaning
The command was not executed because the specifications in one of the TRANSFER-ADMISSION operands were incorrect.

Response
Define the required TRANSFER ADMISSION or correct it.
Repeat the command if necessary.

FTR2047 OPENFT: Request rejected by local FTAC.

Meaning
The command was not executed because the request was rejected by the product openFT-AC due to a lack of authorization.

Response
Use the return code in the logging record to determine and remove the cause. Repeat the command.

FTR2048 OPENFT: Function not supported for protocol '(&00)'.

Meaning
The desired function is not available for the selected protocol.

Response
Select a different protocol.

FTR2049 OPENFT: Remote follow-up processing not supported

Meaning
Remote follow-up processing is only available for the openFT protocol.

Response
Select a different protocol, or specify follow-up processing by means of an FTAC profile.

FTR2050 OPENFT: Data integrity check not supported.

Meaning
The partner system does not support the data integrity check function.

Response
Repeat the request without a file integrity check.

FTR2051 OPENFT: User data encryption not possible for this request.

Meaning
The partner system does not support the data encryption function.

Response
Repeat the request without data encryption or install openFT-CR (or have it installed) on the remote system.

FTR2052 OPENFT: Administration request rejected by remote administration server

Meaning
The command was not executed because the request was rejected by the remote administration server.

Response
Use the return code in the log record on the remote administration server to determine and remove the cause. Repeat the command.

FTR2053 OPENFT: Destination format not supported for transparent transfer

Meaning
The destination file organization parameter is not supported for transparent transfer

Response
Repeat the request without destination file organization parameter.

FTR2054 OPENFT: Invalid command

Meaning
The specified command is not allowed in this context.

Response
Repeat the request with a valid command.

FTR2056 OPENFT: Syntax error in partner name (&00)

Meaning
The syntax of the partner name is wrong.

Response
Correct partner name. Repeat the command.

FTR2058 OPENFT: User data encryption is mandatory

Meaning
The data encryption function is mandatory.

Response
Repeat the request with data encryption.

FTR2070 OPENFT: Request (&00). openFT is no longer authorized to execute requests for this user

FTR2071 OPENFT: Request (&00). User data encryption not installed.

Meaning
The user data encryption function cannot be used unless openFT-CR is installed.

Response
Use openFT-CR.

FTR2072 OPENFT: Request (&00) has been canceled.

Meaning
The FT request was canceled because
– the command NCANCEL was specified, or
– the time specified in NCOPY has been reached.

Follow-up processing has been started for the local system, provided no error occurred. Follow-up processing is started for the remote system once all the resources are allocated. Local errors are indicated by the message FTR0604 at the start of follow-up processing.

FTR2073 OPENFT: Request (&00). Encryption error

Meaning
Encryption not possible.

FTR2074 OPENFT: Request (&00). '(&01)' could not be created (&02).

Meaning
The command was not executed because the file owner and user requesting the creation of a receive file are not the same.

Response
Match the user ID in the receive system's TRANSFER ADMISSION to the ID of the receive file owner. Repeat the command.

FTR2075 OPENFT: Request (&00). Higher-level directory no longer found

FTR2076 OPENFT: Request (&00). I/O error for '(&01)'(&02).

Meaning
The file can no longer be accessed. It may have been deleted during a transfer.

Response
Repeat the request.

FTR2077 OPENFT: Request (&00). File now locked to prevent multiple access.

Meaning
The command was not executed because the send file or the receive file is already locked by another process so that it cannot be simultaneously updated.

Response
Repeat the command later or unlock the file. After a system crash you may need to verify files that are not closed correctly. If the lock is caused by an FT request, it will be released automatically when the request is finished.

FTR2078 OPENFT: Request (&00). '(&01)' no longer available (&02).

Meaning
The command was not executed because the volume for either the send file or the receive file is not mounted, unknown or reserved, the file extends over more than one private disk.

Response
Inform the operator if necessary.
Repeat the command.

FTR2079 OPENFT: Request (&00). '(&01)' no longer found.

Meaning
The local send or receive file can no longer be accessed because, for example, it was deleted during an interruption of the openFT system.

Response
Restore the file.
Repeat the command.

FTR2080 OPENFT: Request (&00). Home directory no longer found (&01)

FTR2081 OPENFT: Request (&00). '(&01)' gets no more space.

Meaning
The command was not executed (any further) executed because
– the permissible storage space on the receive system for the user ID specified in TRANSFER-ADMISSION has been used up, or
– the receive file has already reached the maximum number of allocations.
Take the appropriate action depending on the cause of the error:

Response
delete all files no longer required on the receive system, or
– ask the system administrator to allocate more storage space, or
– remove empty blocks from the send file, or
– reorganize the file so that it requires fewer allocations, or
– increase the receive file's primary/secondary allocation.
If WRITE-MODE=EXTEND-FILE is specified, restore the receive file.
Repeat the command.

FTR2082 OPENFT: Request (&00). File owner no longer known.

Meaning
The command was not executed because the owner of the send file or receive file is not defined on the relevant system or because the file owner and the user who wants to create a receive file are not the same.

Response
Define the file owner, or correct TRANSFER-ADMISSION or FILE-NAME.
Repeat the command.

FTR2083 OPENFT: Request (&00). Pre-/post-processing error(&01).

Meaning
The command executed as part of local pre-/postprocessing returned a result other than OK.

Response
Correct and repeat the command.

FTR2084 OPENFT: Request (&00). Exit code (&01) for pre-/post-processing (&02).

Meaning
The command executed as part of local pre-/postprocessing returned the exit code (&01).

Response
Correct the command using the exit code (&00) and issue it again.

FTR2085 OPENFT: Request (&00). File password no longer valid.

Meaning
The command was not executed because the password for send file or the receive file is missing or incorrect.

Response
Correct the password in the file description or the command.
Repeat the command.

FTR2086 OPENFT: Request (&00). '(&01)' is now read only.

FTR2087 OPENFT: Request (&00). File structure error(&01).

Meaning
The command was executed due to a file structure error.
File structure errors include:
- The attributes of the send file are incomplete.
- The data of the send file is incompatible with its structure attributes.
- The records of the send file are too long.
- If WRITE-MODE=EXTEND-FILE or -e is specified, the send file and receive file have different structures (e.g. fixed-/variable-length records).
- The send file or receive file in a remote BS2000 system is a member of an old LMS library (not PLAM).

Response
Correct the file or file attributes. If WRITE-MODE=EXTEND-FILE or -e is specified, restore the receive file. Repeat the command.

FTR2088 OPENFT: Request (&00). NDMS error (&01).

Meaning
The request was rejected because the partner system currently does not have the resources available to accept requests.

Response
Repeat the request a little later.

FTR2089 OPENFT: Request (&00). Recovery failed (&01).

Meaning
The restart attempts were unsuccessful (for example, a pre-/postprocessing command could not be completed before the termination of openFT).

Response
Repeat the command.

FTR2090 OPENFT: Request (&00). Error in file transfer completion.

Meaning
An error occurred during the final phase of the file transfer. If it was a long transfer, the recipient is advised to check if the file has still been transferred correctly. However, error follow-up processing will be started if it was specified.

Response
Repeat the request, if necessary.

FTR2091 OPENFT: Requests only partially completed; (&00) of (&01) files were transferred

Meaning
In the case of a synchronous send request with wildcards, not all files were successfully transferred.

Response
Transfer unsuccessfully transferred files again.

FTR2092 OPENFT: Request (&00). Access to '(&01)' no longer permissible (&02).

Meaning
The command was not executed because either the send file or the receive file only permits certain access modes (e.g. read only) or because a directory was specified as either the source or destination of a file transfer.

Response
Correct the transfer direction, write mode, file name or file protection attributes.
Repeat the command.

FTR2094 OPENFT: Request (&00). Retention period of file not yet expired.

Meaning
The command was not executed because the retention period protecting the receive file against overwriting has not yet expired (RETENTION PERIOD).

Response
Correct the transfer direction, retention period or file name. Repeat the command.

FTR2095 OPENFT: Request (&00). Extension of file not possible for transparent transfer.

Meaning
The command could not be executed because it is not possible to add to a file in a transparent transfer.

Response
Start transfer without EXTEND.

FTR2096 OPENFT: Request (&00). File structure not supported (&01).

FTR2097 Request (&00). Resulting file name '(&01)' too long(&02)

Meaning
The relative file name was specified in the transfer request. The absolute file name as extended by openFT is longer than permitted.

Response
Shorten the file name or path and repeat the command.

FTR2100 OPENFT: Requests only partially completed; '(&00)' of '(&01)' directories were transferred

Meaning
In the case of a synchronous send request with wildcards, not all directories were successfully transferred.

Response
Transfer unsuccessfully transferred directories again.

FTR2109 OPENFT: Request (&00). Connection setup rejected by local transport system.

FTR2110 OPENFT: Request (&00). Data integrity check indicates an error.

Meaning
The integrity of the data was violated.

FTR2111 OPENFT: Encryption/data integrity check not possible. Encryption switched off.

Meaning
There is no key pair set or the key length was set to 0. Requests can only be carried out without data encryption or a data integrity check.

Response
Repeat the request without data encryption, create a key or set a key length >0.

FTR2112 OPENFT: Request (&00). Data integrity check not supported by partner.

Meaning
The partner system does not support the data integrity check.

Response
Repeat the request without a data integrity check.

FTR2113 OPENFT: Request (&00). User data encryption not possible for this request.

Meaning
The partner system does not support the data encryption function.

Response
Repeat the request without data encryption or install openFT-CR (or have it installed) on the remote system.

FTR2114 OPENFT: Request (&00). Identification of local system rejected by remote system '(&01)'.

Meaning
For security reasons or because of an inconsistency, the partner did not accept the instance identification of the local system (for example, because in a network description file both the instance identification and migration identification %.prozessor.entity occur for different partners).

Response
Ensure that the local identification has been entered correctly on the partner system and has not been assigned to a different partner.

FTR2115 OPENFT: Request (&00). Interrupted by remote system

FTR2116 OPENFT: Local application (&00) not defined

Meaning
The local application is not defined in the transport system.

Response
Make the local application known to the local transport system.

FTR2117 OPENFT: Local application (&00) not available

FTR2118 OPENFT: Request (&00). Authentication of local system failed.

Meaning
The local system could not be authenticated by the partner system.

Response
Give the current public key file to the partner and name it correctly there. Repeat the command.

FTR2119 OPENFT: Request (&00). Local system unknown in remote system.

Meaning
The local system is not known on the partner system (e.g. BS2000/OSD or z/OS).

Response
Make the local system known on the partner system and repeat the command.

FTR2120 OPENFT: Remote system '(&00)' unknown.

Meaning
The partner specified as the remote system cannot be expanded to an address on the local system.

Response
Correct the specification for the partner or add the partner to the partner list and repeat the command.

FTR2121 OPENFT: Request (&00). Authentication of partner failed.

Meaning
The remote system could not be authenticated by the local system.

Response
Get the current public key file from the partner and name it correctly.

FTR2122 OPENFT: Request (&00). FT session rejected or disconnected. Reason (&01)

FTR2123 OPENFT: Request (&00). OSS call error (&01).

Meaning
The command was not executed because the session instance detected a communication error.
(&00): error code.

Response
Take the appropriate action in accordance with the error code.

FTR2124 OPENFT: Request (&00). No free connection

Meaning
No more transfers are possible because the maximum number of simultaneous transfers has been reached.

Response
Check whether the transport system is working (or have it checked).

FTR2125 OPENFT: Request (&00). Connection lost.

Meaning
No data transfer took place because of a line interrupt or a line protocol error.

Response
Repeat the request.

FTR2126 OPENFT: Request (&00). Transport system error. Error code (&01)

Meaning
An error occurred in the transport system during processing of a FTSTART command or a file transfer or file management request.

Response
Take the appropriate action in accordance with the error code. Most often the occurrence of this message indicates that the partner addressed is not known to the transport system.

FTR2127 OPENFT: Request (&00). No data traffic within (&01) seconds

Meaning
No data transfer took place within the period of seconds specified because, for example, the connection is interrupted, the partner is not sending and the local system is waiting for data.

Response
Repeat the request.

FTR2129 OPENFT: No further files will be transferred

Meaning
Check the previous transfer error code.

FTR2140 OPENFT: Request (&00). Remote system: openFT is not authorized to execute requests for this user.

FTR2141 OPENFT: Request (&00). Remote system: Directory (&01) is not empty

Meaning
The command could not be executed because there are files in the specified directory of the partner system.

Response
Delete all the files in the directory first and repeat the command.

FTR2142 OPENFT: Request (&00). Remote system: File attributes do not match the request parameters (&01)

Meaning
The command could not be executed because the file attributes on the remote system do not agree with the request parameters (e.g. a directory was specified instead of a remote file).

Response
Check the file name on the remote system and correct it. Repeat the command.

FTR2143 OPENFT: Request (&00). Remote system: Attributes could not be modified (&01).

Meaning
The properties of the file could not be modified as desired in the command.
Possible reasons are for the remote file:
– No access rights to the file.
– The combination of access rights required is not supported by the remote system.
– If the remote system is a BS2000: the file is protected by ACL.

FTR2144 OPENFT: Request (&00). Remote system: File/directory (&01) could not be created (&02)

Meaning
The command was not executed because the file owner and user requesting the creation of a receive file are not the same.

Response
Match the user ID in the receive system's TRANSFER-ADMISSION to the ID of the receive file owner. Repeat the command.

FTR2145 OPENFT: Request (&00). Remote system: CCS name unknown or not supported.

Meaning
The request could not be completed because the CCS is unknown in the partner system.

FTR2146 OPENFT: Request (&00). Remote system: Higher-level directory not found

Meaning
The command was not executed because the higher-level directory could not be found on the partner system.

Response
Create the directory on the remote system or correct the remote directory name and repeat the command.

FTR2147 OPENFT: Request (&00). Remote system: File/directory '(&01)' already exists.

Meaning
The command was not executed. Possible reasons:
– The command was not executed because an existing receive file cannot be created with 'WRITE-MODE=NEW' or the -n option. WRITE-MODE=NEW or -n may also have been set by a restriction in the access authorization used.
– ftcredir: The specified directory already exists.

Response
Either delete the receive file before repeating the command or reenter the command specifying WRITE-MODE=REPLACE-FILE or using different access authorization.

FTR2148 OPENFT: Request (&00). Remote system: Transfer of file generation groups not supported.

Meaning
The command was not executed because the FT system can only transfer single file generations.

Response
Repeat the command using the name of a single file generation.

FTR2149 OPENFT: Request (&00). Remote system: Access error for '(&01)' (&02).

Meaning
(&02): DMS error, possibly the transfer ID. The FT system continues to run after output of the message.

Response
Take the appropriate action in accordance with the error code.

FTR2150 OPENFT: Request (&00). Remote system: Resulting file name too long (&01).

Meaning
A syntax error other than 'operand missing' (FTR0010) or 'keyword unknown' (FTR0011) has been detected. Possible reasons:
– Values assigned outside the valid range
– Invalid operand separators
– Invalid value assignment characters
– Partially qualified file names

Response
Repeat the command using the correct syntax.

FTR2151 OPENFT: Request (&00). Remote system: File locked to prevent multiple access.

Meaning
The command was not executed because either the send file or the receive file is already locked by another process to prevent it from being updated simultaneously.

Response
Repeat the command later or unlock the file on the remote system. After a system crash in BS2000 you may need to verify files not closed correctly. If the lock is caused by an FT request, it will be released automatically when the request is finished.

FTR2152 OPENFT: Request (&00). Remote system: No file or directory name specified.

Meaning
The command was not executed because the file ID was neither specified explicitly nor by the 'TRANSFER-ADMISSION' used.

Response
Repeat the command, specifying the file ID explicitly or using a TRANSFER ADMISSION that defines the file ID.

FTR2153 OPENFT: Request (&00). Remote system: Invalid management password.

FTR2154 OPENFT: Request (&00). Remote system: File/directory '(&01)' not available (&02).

Meaning
The command was not executed because the volume for either the send file or the receive file is not mounted, unknown or reserved, the file extends over more than one private disk, or an attempt has been made to transfer a file migrated by HSM without specifying the remote transfer admission.

Response
Inform the operator if necessary or carry out an HSM recall for the file or specify the remote transfer admission. Repeat the command.

FTR2155 OPENFT: Request (&00). Remote system: File/directory '(&01)' not found.

Meaning
The command was not executed because the send file is not or no longer in the catalog or on a volume of the remote system.

Response
Correct the remote file name, read the file in from tape or restore the send file.
Repeat the command.

FTR2156 OPENFT: Request (&00). Remote system: Home directory not found (&01)

FTR2157 OPENFT: Request (&00). Remote system: Renaming not possible (&01)

FTR2158 OPENFT: Request (&00). Remote system: Not enough space for '(&01).

Meaning
The command was not executed (any further) because the permissible storage space on the receive system for the user ID specified in TRANSFER-ADMISSION has been used up. The receive file is no longer created/extended after the problem has occurred.

Response
Take the appropriate action depending on the cause of the error:
– delete all files no longer required on the receive system, or
– ask the system administrator to allocate more storage space, or
– increase the receive file's primary/secondary allocation.

If WRITE-MODE=EXTEND-FILE is specified, restore the receive file.
Repeat the command.

FTR2159 OPENFT: Request (&00). Remote system: File owner unknown.

Meaning
The command was not executed because the owner of either the send file or the receive file was not defined on the relevant system or because the file owner and the user requesting the creation of a receive file are not the same.

Response
Define the file owner, correct TRANSFER-ADMISSION or FILE-NAME.
Repeat the command.

FTR2160 OPENFT: Request (&00). Remote system: Invalid file password.

Meaning
The command was not executed because the password for the send file or the receive file is missing or incorrect.

Meaning
Correct the password in the file description or the command. Repeat the command.

FTR2161 OPENFT: Request (&00). Remote system: Retention period of file not yet expired.

Meaning
The command was not executed because the retention period protecting the receive file against overwriting has not yet expired.

Response
Correct the transfer direction, retention period or file name. Repeat the command.

FTR2162 OPENFT: Request (&00). Remote system: File/directory '(&01)' is read only.

Meaning
The file or directory is write-protected.

Response
Correct the remote file name or remove the write protection of the remote file.
Repeat the command.

FTR2163 OPENFT: Request (&00). Remote system: File structure not supported(&01).

Meaning
The request cannot be carried out because the file structure is not supported. For example, an attempt was made to get a PLAM library or ISAM file from the BS2000 system.

Response
Transfer the file transparently.

FTR2164 OPENFT: Request (&00). Remote system: Syntax error in resulting file name(&01).

Meaning
A syntax error other than 'operand missing' (FTR0010) or 'keyword unknown' (FTR0011) has been detected.
Possible reasons:
– Values assigned outside the valid range
– Invalid operand separators
– Invalid value assignment characters
– Partially qualified file names

Meaning
Repeat the command using the correct syntax.

FTR2165 OPENFT: Request (&00). Remote system: Transparent file transfer not supported.

Meaning
The request could not be carried out because the partner system does not support the transfer of files in a transparent format.

FTR2166 OPENFT: Request (&00). Remote system: Extension of file not possible for transparent transfer.

Meaning
The command could not be executed because it is not possible to add to a file in a transparent transfer.

FTR2167 OPENFT: Request (&00). Remote system: Access to '(&01)' denied (&02).

Meaning
The command was not executed because the remote file only permits certain access modes.

Response
Correct the transfer direction, file name or file protection attributes on the remote system. Repeat the command.

FTR2168 OPENFT: Request (&00). Remote system: Follow-up processing exceeds length limit.

Meaning
The maximum length of follow-up processing was exceeded; see the command syntax description.

Response
Shorten the follow-up processing, or use procedures. Repeat the command.

FTR2169 OPENFT: Request (&00). Remote system: Transfer admission invalid.

Meaning
The command was not executed because the specifications in one of the TRANSFER-ADMISSION operands are incorrect or the request was rejected by FTAC because of insufficient authorization.

Response
Define the requisite TRANSFER-ADMISSION or correct it or check the authorization entered in FTAC. Repeat the command if necessary.

FTR2170 OPENFT: Request (&00). Remote system: Function not supported (&01).

FTR2171 OPENFT: Request (&00). Remote system: Processing admission invalid.

Meaning
The command was not executed because the specifications in one of the PROCESSING-ADMISSION operands are incorrect.

Response
Define the required PROCESSING ADMISSION or correct it.
Repeat the command if necessary..

FTR2172 OPENFT: Request (&00). Remote system: Request queue full.

Meaning
The command was not executed because the maximum number of permissible file transfer requests has been reached.

Response
Notify the FT administrator. Repeat the command later.

FTR2173 OPENFT: Request (&00). Remote system: User data encryption is mandatory

Meaning
The remote system only accepts requests using data encryption.

Response
Repeat the request using data encryption.

FTR2174 OPENFT: No files corresponding to specified pattern found

Meaning
Specify the correct file name pattern.

FTR2195 OPENFT: Request (&00). Remote system: openFT is not longer authorized to execute requests for this user.

FTR2196 OPENFT: Request (&00) has been canceled in the remote system.

Meaning
The request was deleted on the remote system before termination.

FTR2197 OPENFT: Request (&00). Remote system: File/directory '(&01)' could not be created(&02).

Meaning
The command was not executed because the file owner and user requesting the creation of a receive file are not the same.

Response
Match the user ID in the receive system's TRANSFER-ADMISSION to the ID of the receive file owner. Repeat the command.

FTR2198 OPENFT: Request (&00). Remote system: Higher-level directory no longer found

FTR2199 OPENFT: Request (&00). Remote system: I/O error for '(&01)' (&02).

Meaning
An error occurred at input/output. Possible cause:
– BS2000: DMS error, possibly the transfer ID.
– The send or receive files was deleted during transfer.
The FT system continues to run after the message has been issued.

Response
Take the appropriate action in accordance with the error code.

FTR2200 OPENFT: Request (&00). Remote system: File now locked to prevent multiple access.

Meaning
The command was not executed because either the send file or the receive file is already locked by another process to prevent it from being updated simultaneously. An attempt is made, for example, to access a library opened in z/OS.

Response
Repeat the command later or unlock the file. After a system crash you may need to verify files not closed correctly. If a lock is caused by an FT request, it will be released automatically when the request is finished.

FTR2201 OPENFT: Request (&00). Remote system: File/directory '(&01)' no longer available(&02).

Meaning
The command was not executed because the volume for either the send file or the receive file is not mounted, unknown or reserved, or because the file extends over more than one private disk or an attempt has been to transfer a file migrated by HSM.

Response
Inform the operator if necessary or carry out an HSM recall for the file.
Repeat the command.

FTR2202 OPENFT: Request (&00). Remote system: File/directory '(&01)' no longer found.

Meaning
The command was not executed because the remote file is not or no longer in the catalog or on a volume of the corresponding system (e.g. after a restart).

Response
Restore the remote file. Repeat the command.

FTR2203 OPENFT: Request (&00). Remote system: Home directory no longer found (&01)

FTR2204 OPENFT: Request (&00). Remote system: File/directory '(&01)' gets no more space.

Meaning
The command was not executed (any further) because
– the permissible storage space on the receive system for the user ID specified in TRANSFER-ADMISSION has been used up, or
– the send file contains too long a sequence of empty blocks, or
– the primary and/or secondary allocation of the password-protected receive file is too small.

The receive file can no longer be created/extended after the problem occurs.

Response
Take the appropriate action depending on the cause of the error:
– delete all files no longer required on the receive system, or
– ask the system administrator to allocate more storage space, or
– remove empty blocks from the send file, or
– increase the receive file's primary/secondary allocation.
If WRITE-MODE=EXTEND-FILE is specified, restore the receive file.
Repeat the command.

FTR2205 OPENFT: Request (&00). Remote system: File owner no longer known.

Meaning
The command was not executed because the owner of either the send file or the receive file is not defined on the relevant system, or because the file owner and the user requesting the creation of the receive file are not the same.

Response
Define the file owner, correct TRANSFER-ADMISSION or FILE-NAME.
Repeat the command.

FTR2206 OPENFT: Request (&00). Remote system: Pre-/post-processing error (&01).

Meaning
The command executed in local pre-/postprocessing returned a result value other than OK.

Response
Correct the pre-/postprocessing command and issue it again.

FTR2207 OPENFT: Request (&00). Remote system: Exit code (&01) during pre-/post-processing (&02).

Meaning
The command executed in local pre-/postprocessing returned the exit code (&01).

Response
Correct the pre-/postprocessing command in accordance with the exit code and issue it again.

FTR2208 OPENFT: Request (&00). Remote system: File password no longer valid.

Meaning
The command was not executed because the password for the send file or receive file is missing or incorrect.

Response
Correct the password in the file description or the command. Repeat the command.

FTR2209 OPENFT: Request (&00). Remote system: File/directory '(&01)' is now read only.

FTR2210 OPENFT: Request (&00). Remote system: File structure error (&01).

Meaning
The command was not executed due to a file structure error.
File structure errors include:
– The attributes of the send file are incomplete.
– The data of the send file is incompatible with its structure attributes.
– The records of the send file are too long.
– If WRITE-MODE=EXTEND-FILE or the -e parameter are specified, the send file and receive file have different structures (e.g. fixed-/variable-length records).
– BS2000: The send or receive file is a member of an old LMS library (not PLAM).
– BS2000: The send file has an odd block factor (e.g. BLKSIZE=(STD,1)), and the receive file is stored on an NK4 pubset.

Response
Correct the file or file attributes. If WRITE-MODE=EXTEND-FILE is specified, restore the receive file. Repeat the command.

FTR2211 OPENFT: Request (&00). Remote system: NDMS error (&01).

Response
Repeat the request a little later.

FTR2212 OPENFT: Request (&00). Recovery failed (&01).

Meaning
The restart could not be carried out. It may not have been possible to complete restart-capable pre-/postprocessing before termination of the server process (waiting time: max. minutes).

Response
Repeat the command.

FTR2213 OPENFT: Request (&00). Remote system: Resource bottleneck.

Meaning
The order was rejected because the partner system currently does not have the resources available to accept requests.

Response
Repeat the request a little later.

FTR2214 OPENFT: Request (&00). Remote system: Access to '(&01)' is no longer permissible(&02).

Meaning
The command was not executed because
– the send file or receive file only permits certain access modes (e.g. read only) or a directory was specified as the source or destination of a file transfer.
– or because no valid password for an FTAC profile has been stored in the local system for executing the ftexec command from a remote system.

Response
Correct the transfer direction, write mode, file name or file protection attributes or specify a valid password for the FTAC profile. Repeat the command.

FTR2216 OPENFT: Request (&00). Remote system: File structure not supported (&01).

Meaning
The request cannot be carried out because the file structure is not supported. An attempt was made, for example, to get a PLAM library or ISAM file from BS2000.

Response
Transfer the file transparently.

FTR2217 OPENFT: Request (&00). Remote system: Retention period of file not yet expired.

Meaning
The command was not executed because the retention period protecting the receive file against overwriting has not yet expired.

Response
Correct the transfer direction, retention period or file name. Repeat the command.

FTR2218 OPENFT: Request (&00). Remote system: Extension of file not possible for transparent transfer.

Meaning
The command could not be executed because it is not possible to add to a file in a transparent transfer.

FTR2225 OPENFT: Information output canceled.

Meaning
A show command was interrupted, for example.

Response
Repeat the command.

6.5.3 FTC messages

FTC0001 FTAC VERSION (&OO) ACTIVE

Meaning
FTAC initialization is concluded.

FTC0003 (&OO) LOGGING RECORDS DELETED

Meaning
The specified number of records have been deleted from the logging file.

FTC0050 CMD ACCEPTED. WARNING: LOWER ADM-LEVEL REMAINS IN EFFECT

Meaning
The set security level exceeds the administrator's limit value and will remain without effect until the administrator's limit value is increased.

Response
Request a higher maximum security level from the FTAC administrator.

FTC0051 CMD ACCEPTED. WARNING: TRANSFER-ADMISSION EXISTS AS USER ID

Meaning
A user ID with the same name already exists in the system.

Response
The message is simply intended to indicate a possible confusion.

FTC0052 CMD TERMINATED. INFORMATION INCOMPLETE

Meaning
Information output has been interrupted.

Response
Repeat the command if necessary.

FTC0053 CMD TERMINATED. NO FT PROFILE FOUND

Meaning
There is no FT profile for the specified criteria.

FTC0054 CMD ACCEPTED. NO INFORMATION AVAILABLE

Meaning
There is no information on the specified criteria.

FTC0055 WARNING: PARTNER RESTRICTION DOES NOT LONGER EXIST

FTC0056 WARNING: TRANSFER ADMISSION LOCKED

FTC0057 WARNING: ATTRIBUTES OF TRANSFER ADMISSION ARE IGNORED

Meaning
In the case of a profile with transfer admission *NOT-SPECIFIED, VALID, USAGE and EXPIRATION-DATE are ignored.

FTC0070 CMD TERMINATED. SHORTAGE OF RESOURCES

Meaning
The command cannot be executed due to a lack of resources.

Response
Repeat the command.

FTC0071 CMD REJECTED. OPENFT NOT ACTIVE

Meaning
openFT has not been activated, FTAC is therefore inactive.

Response
Ask the system administrator to activate openFT. FTAC will be activated by openFT.

FTC0100 CMD REJECTED. FT PROFILE ALREADY EXISTS

Meaning
There is already an FT profile with the specified name.

Response
Select another name.

FTC0101 CMD REJECTED. TRANSFER ADMISSION ALREADY EXISTS

Meaning
There is already an FT profile with the specified transfer admission.

Response
You should choose the TRANSFER-ADMISSION more carefully to ensure greater security.

FTC0102 FILE ALREADY EXISTS

FTC0103 INVALID FILE CONTENT OR ACCESS TO FILE DENIED

Meaning
The file is not an FTAC export file or access is prohibited.

FTC0104 ACCESS TO USER ID DENIED OR USER ID DOES NOT EXIST.

Meaning
Access to user ID rejected.
The user ID does not exist.

FTC0105 ACCESS TO FILE DENIED

FTC0106 ACCESS TO TEMPORARY FILE DENIED

FTC0107 NO SPACE AVAILABLE

FTC0108 THE VERSION OF EXPORT FILE IS NOT COMPATIBLE WITH CURRENT VERSION

FTC0109 FILE IS NO FTAC EXPORT FILE

FTC0110 FILE NAME TOO LONG

FTC0111 SYNTAX ERROR IN FILE NAME

FTC0112 CMD REJECTED. EXPIRATION DATE NOT VALID

Meaning
The value of the parameter EXPIRATION-DATE must be between 1970-01-02 and 2038-01-19.

FTC0150 CMD REJECTED. USER NOT AUTHORIZED FOR FTAC COMMANDS

Meaning
There is no password for the admission.

Response
Specify the FTAC password.

FTC0151 CMD REJECTED. USER NOT AUTHORIZED FOR THIS MODIFICATION

Meaning
Only the administrator or owner can perform the modification.

FTC0152 CMD REJECTED. USER NOT AUTHORIZED FOR OTHER USER IDS

Meaning
The specified user ID is not your own user ID.

FTC0153 CMD REJECTED. USER NOT AUTHORIZED FOR OTHER OWNER IDS

Meaning
The specified owner identification is not your own user ID.

FTC0154 CMD REJECTED. NO AUTHORIZATION FOR DELETION OF LOG RECORDS

FTC0155 CMD REJECTED. USER NOT AUTHORIZED FOR DIAGNOSE

Meaning
Only the FT administrator and FTAC administrator may call the diagnostic function.

FTC0156 COMMAND ALLOWED FOR FTAC ADMINISTRATOR ONLY

FTC0157 CMD REJECTED. NO AUTHORIZATION FOR THIS SET OF PARAMETERS

Meaning
The FTAC administrator can only create profiles with a transfer admission specification if he or she knows the complete user ID or possesses the SU privilege.

Response
Specify the full user ID in the form user-adm=(uid,acc,pw).

FTC0170 CMD REJECTED. GIVEN PARTNER UNKNOWN

Meaning
The specified partner is unknown within the group of partner systems permitted for this user.

FTC0171 CMD REJECTED. GIVEN FT PROFILE NAME UNKNOWN

Meaning
The specified profile does not exist.

FTC0172 CMD REJECTED. INVALID USER ADMISSION

Meaning
The specified user admission does not exist in the system.

Response
The USER-IDENTIFICATION, ACCOUNT or PASSWORD is incorrect.

FTC0173 CMD REJECTED. INVALID PROCESSING ADMISSION

Meaning
The specified processing admission does not exist in the system.

Response
The USER-IDENTIFICATION, ACCOUNT or PASSWORD specification is incorrect.

FTC0174 CMD REJECTED. MODIFICATION INVALID FOR NOT UNIQUE SELECTION CRITERIA

Meaning
The parameters "NEW-NAME" and "TRANSFER-ADMISSION" may only be used in combination with unique selection criteria ("NAME" or "TRANSFER-ADMISSION").

Response
Choose a unique selection criterion.

FTC0175 CMD REJECTED. MODIFICATION INVALID FOR STANDARD AUTHORIZATION RECORD

Meaning
The parameter "NEW-PASSWORD" may not be specified for *STD.

FTC0176 CMD REJECTED. GIVEN USER ID UNKNOWN

Meaning
The specified user ID does not exist in the system.

FTC0177 FILE UNKNOWN

FTC0178 MULTIPLE PARTNER NAME SPECIFIED

FTC0179 VIOLATION OF MAXIMAL NUMBER OF PARTNER RESTRICTIONS

FTC0180 MULTIPLE USERID SPECIFIED

FTC0181 MULTIPLE FT PROFILE NAME SPECIFIED

FTC0182 TOTAL MAXIMUM PARTNER NAME LENGTH EXCEEDED

Meaning
The total length of the partner names may not exceed 1000 characters.

FTC0183 CMD REJECTED. PARTNER NOT SUPPORTED

FTC0184 Invalid parameter transfer admission for profile *STD

Meaning
The transfer admission of the default profile must be *NOT-SPECIFIED.

FTC0185 COMBINATION OF THESE TRANSFER FUNCTIONS NOT ALLOWED

FTC0200 CMD REJECTED. FOLLOW-UP PROCESSING TOO LONG

Meaning
The total length of the two follow-up processing commands is too great.

Response
Use shorter commands (e.g. by using procedures).

FTC0201 USER ID TOO LONG

FTC0202 PROFILE NAME TOO LONG

FTC0203 TRANSFER ADMISSION TOO LONG

FTC0204 PARTNER TOO LONG

FTC0205 FULLY QUALIFIED FILE NAME TOO LONG

FTC0206 PARTIALLY QUALIFIED FILE NAME TOO LONG

FTC0207 PROCESSING COMMAND TOO LONG

FTC0208 INVALID DATE SPECIFIED

FTC0209 INVALID TIME SPECIFIED

FTC0210 TRANSFER ADMISSION TOO SHORT

FTC0211 PARAMETERS (&00) AND (&01) MAY NOT BE SPECIFIED TOGETHER

FTC0212 LICENSE CHECK ERROR (&00) FOR FTAC

FTC0213 MANDATORY PARAMETER PROFILE NAME IS MISSING

FTC0214 MANDATORY PARAMETER FILE NAME IS MISSING

FTC0215 SYNTAX ERROR IN PARAMETER (&OO)

FTC0216 PASSWORD TOO LONG

FTC0217 TEXT TOO LONG

FTC0218 TOO MANY PARTNERS

FTC0219 TOO MANY USERS

FTC0220 TOO MANY PROFILES

FTC0250 LOAD ERROR. ERROR-CODE (&OO)

FTC0251 CMD REJECTED. FTAC NOT AVAILABLE

Meaning
openFT-AC has not been installed completely.

Response
The system administrator must check the openFT-AC installation.

FTC0253 FTAC COMMAND NOT FOUND IN SYNTAXFILE

Meaning
The openFT-AC syntax file has been merged incorrectly or incompletely into the system syntax file.

Response
The system administrator must check the system syntax file.

FTC0254 SYSTEM ERROR. ERRORCODE (&OO)

Meaning
A system error has occurred.

Response
Generate diagnostic material and inform the staff responsible for system diagnostics.

FTC0255 CMD TERMINATED. SYSTEM ERROR

Meaning
A system error has occurred.

Response
Inform the system administrator. At the same time a message is issued to the operator terminal providing exact troubleshooting information.

FTC1001 SUBMISSION REJECTED. INVALID TRANSFER-ADMISSION

Meaning
The specified TRANSFER-ADMISSION is not defined in any FT profile.

FTC1002 SUBMISSION REJECTED. INVALID INITIATOR

Meaning
The FT profile restricts initiatives to LOCAL or REMOTE.

FTC1003 SUBMISSION REJECTED. INVALID TRANSFER-DIRECTION

Meaning
The FT profile restricts the TRANSFER-DIRECTION to TO or FROM.

FTC1004 SUBMISSION REJECTED. INVALID PARTNER NAME

Meaning
The FT profile does not permit any requests involving the specified partner system.

FTC1005 SUBMISSION REJECTED. VIOLATION OF MAX-PARTNER-LEVEL

Meaning
The partner system's security level exceeds the value specified for MAX-PARTNER-LEVEL in the FT profile.

FTC1006 SUBMISSION REJECTED. SYNTAX ERROR OF FILE NAME EXPANSION

Meaning
The FT profile does not permit the specification of a file name or file name expansion in the request.

FTC1007 SUBMISSION REJECTED. VIOLATION OF LIBRARY RESTRICTION

Meaning
The file or library name specified in the command infringes the LIBRARY restriction in the profile.

FTC1008 SUBMISSION REJECTED. VIOLATION OF ELEMENT RESTRICTION

Meaning
The FT profile does not permit the specification ELEMENT in the request.

FTC1009 SUBMISSION REJECTED. VIOLATION OF ELEMENT-VERSION RESTRICTION

Meaning
The FT profile does not permit the specification ELEMENT-VERSION in the request.

FTC100A SUBMISSION REJECTED. VIOLATION OF ELEMENT-TYPE RESTRICTION

Meaning
The FT profile does not permit the specification ELEMENT-TYPE in the request.

FTC100B SUBMISSION REJECTED. VIOLATION OF FILE-PASSWORD RESTRICTION

Meaning
The FT profile does not permit the specification FILE-PASSWORD in the request.

FTC100C SUBMISSION REJECTED. VIOLATION OF USER-IDENTIFICATION(PROCESSING-ADMISSION) RESTRICTION

Meaning
The FT profile does not permit the specification USER-IDENTIFICATION in the request's PROCESSING-ADMISSION.

FTC100D SUBMISSION REJECTED. VIOLATION OF ACCOUNT(PROCESSING-ADMISSION) RESTRICTION

Meaning
The FT profile does not permit the specification ACCOUNT in the request's PROCESSING-ADMISSION.

FTC100E SUBMISSION REJECTED. VIOLATION OF PASSWORD(PROCESSING-ADMISSION) RESTRICTION

Meaning
The FT profile does not permit the specification PASSWORD in the request's PROCESSING-ADMISSION.

FTC100F SUBMISSION REJECTED. VIOLATION OF SUCCESS-PROCESSING RESTRICTION

Meaning
The FT profile does not permit the specification SUCCESS-PROCESSING.

FTC1010 SUBMISSION REJECTED. VIOLATION OF FAILURE-PROCESSING RESTRICTION

Meaning
The FT profile does not permit the specification FAILURE-PROCESSING.

FTC1011 SUBMISSION REJECTED. VIOLATION OF WRITE-MODE RESTRICTION

Meaning
The FT profile does not permit the specified WRITE-MODE.

FTC1012 SUBMISSION REJECTED. INVALID FT-FUNCTION

Meaning
The FT profile does not permit the desired FT function.

FTC1013 SUBMISSION REJECTED. VIOLATION OF PROFILE WITH CHIPCARD-ID

Meaning
The profile may only be used with a chipcard.

FTC1014 SUBMISSION REJECTED. VIOLATION OF DATA ENCRYPTION RESTRICTION

Meaning
The profile does not permit the value DATA-ENCRYPTION in the request.

FTC2001 SUBMISSION REJECTED. SYNTAX ERROR ON FILE NAME EXPANSION

Meaning
The combination of the FT profile's FILE-NAME and FILE-NAME expansion resulted in a syntax error.

FTC2002 SUBMISSION REJECTED. SYNTAX ERROR ON LIBRARY NAME EXPANSION

Meaning
The combination of the FT profile's LIBRARY name and LIBRARY expansion resulted in a syntax error.

FTC2003 SUBMISSION REJECTED. SYNTAX ERROR ON ELEMENT NAME EXPANSION

Meaning
The combination of the FT profile's ELEMENT name and ELEMENT expansion resulted in a syntax error.

FTC2004 SUBMISSION REJECTED. TOTAL LENGTH OF RESULT PROCESSING EXCEEDS 500 CHARACTERS

Meaning
SUCCESS and FAILURE processing including the expansions defined in the FT profile exceeds 1000 characters.

FTC3001 SUBMISSION REJECTED. INVALID USER-IDENTIFICATION

Meaning
The TRANSFER-ADMISSION's USER-IDENTIFICATION or, if an FT profile is used, the USER-ADMISSION is invalid.

FTC3002 SUBMISSION REJECTED. INVALID ACCOUNT

Meaning
The TRANSFER-ADMISSION's ACCOUNT specification or, if an FT profile is used, the USER-ADMISSION is invalid.

FTC3003 SUBMISSION REJECTED. INVALID PASSWORD

Meaning
The TRANSFER-ADMISSION's PASSWORD specification or, if an FT profile is used, the USER-ADMISSION is invalid.

FTC3004 SUBMISSION REJECTED. TRANSFER ADMISSION LOCKED

Meaning
The transfer admission is locked. The reasons may be ascertained from the output from the FTSHWPRF command.

FTC3011 SUBMISSION REJECTED. VIOLATION OF USER OUTBOUND SEND LEVEL

Meaning
The partner system's security level is not permitted by the user for the OUTBOUND SEND function class.

FTC3012 SUBMISSION REJECTED. VIOLATION OF USER OUTBOUND RECEIVE LEVEL

Meaning
The partner system's security level is not permitted by the user for the OUTBOUND RECEIVE function class.

FTC3013 SUBMISSION REJECTED. VIOLATION OF USER INBOUND SEND LEVEL

Meaning
The partner system's security level is not permitted by the user for the INBOUND SEND function class.

FTC3014 SUBMISSION REJECTED. VIOLATION OF USER INBOUND RECEIVE LEVEL

Meaning
The partner system's security level is not permitted by the user for the INBOUND RECEIVE function class.

FTC3015 SUBMISSION REJECTED. VIOLATION OF USER INBOUND PROCESSING LEVEL

Meaning
The partner system's security level is not permitted by the user for the INBOUND PROCESSING function class.

FTC3016 SUBMISSION REJECTED. VIOLATION OF USER INBOUND FILE MANAGEMENT LEVEL

Meaning
The partner system's security level is not permitted by the user for the INBOUND FILE MANAGEMENT function class.

FTC3021 SUBMISSION REJECTED. VIOLATION OF ADM OUTBOUND SEND LEVEL

Meaning
The partner system's security level is not permitted by the administrator for the OUTBOUND SEND function class.

FTC3022 SUBMISSION REJECTED. VIOLATION OF ADM OUTBOUND RECEIVE LEVEL

Meaning
The partner system's security level is not permitted by the administrator for the OUTBOUND RECEIVE function class.

FTC3023 SUBMISSION REJECTED. VIOLATION OF ADM INBOUND SEND LEVEL

Meaning
The partner system's security level is not permitted by the administrator for the INBOUND SEND function class.

FTC3024 SUBMISSION REJECTED. VIOLATION OF ADM INBOUND RECEIVE LEVEL

Meaning
The partner system's security level is not permitted by the administrator for the INBOUND RECEIVE function class.

FTC3025 SUBMISSION REJECTED. VIOLATION OF ADM INBOUND PROCESSING LEVEL

Meaning
The partner system's security level is not permitted by the administrator for the INBOUND PROCESSING function class.

FTC3026 SUBMISSION REJECTED. VIOLATION OF ADM INBOUND FILE MANAGEMENT LEVEL

Meaning
The partner system’s security level is not permitted by the administrator for the INBOUND FILE MANAGEMENT function class.

6.6 Using openFT in z/OS systems without the TSO interactive system

openFT is intended for use under the z/OS operating system. The commands are passed to the TSO command processor. Nevertheless,

openFT can also be used without the TSO interactive system. In this case, the IBM utility IKJEFT01 must be used to call the TSO command processor in batch mode.

In order to be able to work with openFT without the TSO interactive system, all commands must be included in batch jobs. These jobs are initiated via the IBM utility IEBGENER. IEBGENER reads the job information from a file and passes it on to the Job Entry Subsystem (JES2/3).

It is not then possible to set the openFT installation parameters using the FJGEN command (see) within a TSO dialog. Instead, the installation parameters must be set using a parameter library (see section “Setting up the FT parameter library” on page 57).

Issuing TSO commands

These commands are processed by the TSO command processor. In an exclusive z/OS batch environment, the IKJEFT01 utility provides the appropriate interface.

Example of a batch job including the NCOPY command:

```
//USERN      JOB    .....
//NCOPY      EXEC PGM=IKJEFT01
//SYSPRINT   DD   SYSOUT=*
//SYSTSPRT   DD   SYSOUT=*
//SYSTSIN    DD   *
NCOPY TRANS=TO,PARTNER=MVS2,+
LOC=(FILE=......
...
...
/*
//
```

Index