English

FUJITSU Software

openFT (BS2000) 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

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

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

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

- FTAC:

  FTAC provides extended system and data access protection. FTAC stands for File Transfer Access Control.
  On BS2000 systems, FTAC is provided by the add-on product openFT-AC.

- openFT-FTAM:

  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 BS2000 system.

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*.

You will also find the Readme files on the Softbook DVD.

Information under BS2000 systems

When a Readme file exists for a product version, you will find the following file on the BS2000 system:

`SYSRME.<product>.<version>.<lang>`

This file contains brief information on the Readme file in English or German (<lang>=E/D). You can view this information on screen using the `/SHOW-FILE` command or an editor. The `/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=<product>` command shows the user ID under which the product's files are stored.

Additional product information

Current information, version and hardware dependencies, and instructions for installing and using a product version are contained in the associated Release Notice. These Release Notices are available online 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 startup

This chapter describes the actions and preconditions required to install and run openFT and any of the optional components openFT-FTAM, openFT-AC, openFT-FTP and openFT-CR in BS2000.

2.1 Installing openFT

openFT V12.1 requires the following software

– BS2000/OSD as of V10.0
– openNet server as of V3.5 (i.e. BCAM ≥ V22.0)

The following versions are required when using the optional add-on components:

– openFT-FTAM V12.1
– openFT-AC V12.1
– openFT-CR V12.1
– openFT-FTP V12.1

If you want to make use of the POSIX functionality, you will also need the BS2000 component POSIX.

openFT-FTAM V12.1 requires:

– openFT ≥ V12.1
– OSS ≥ V4.1D

openFT-AC, openFT-FTP and openFT-CR V12.1 require:

– openFT ≥ V12.1

Delivery of openFT is done via the software delivery and information system SOLIS2. Installation is done via IMON. The installation routine incorporates the required BS2000-specific tasks such as the MSGFILE update, subsystem catalog entries and the integration of the SDF syntax file.

Whenever a program which uses FT interfaces is compiled, the file SYSLIB.OPENFT.121 for COBOL and ASSEMBLER programs must be available. This file must be available as a shareable file in the system, but need not be located under the TSOS ID.

2.1.1 Initial installation of openFT (BS2000)

openFT is a subsystem and is not generated when the BS2000 system is generated.

The FT administrator commands can be issued from the console. Administration from the terminal requires the FT-ADMINISTRATION privilege, assigned by default to TSOS. If SECOS is in use, this privilege can be assigned to other user IDs. See the SECOS manual for details.

In order to ensure the usability of the COBOL program interface, the file SYSRTC.FT (runtime module for the COBOL program interface) must be shareable under the SYSFJAM ID (`SHARE=YES, ACCESS=READ`). COBOL programs produced with the COBOL interface load the runtime module from this ID.

The product PAMINT is used to convert from keyed to nonkey files (and vice versa). This product belongs to the BS2000 basic configuration and must be available under TSOS.

2.1.2 Version change and compatibility

Compatibility

openFT V12.1 is compatible with openFT V12.0, i.e. all V12.0 functions can be used unchanged and without restriction in V12.1.

openFT V12.1 is no longer supported on SX servers.

Change of version

Before performing installation with IMON, the following activities must be performed in all the instances present:

1. Back up the operating parameter settings, the partner list entries and, if applicable, the FTAC environment in procedure files, for details, see section “Backing up the configuration data” on page 78.
2. Back up the following openFT system files to the instance ID (standard instance: $SYSFJAM) in case you should need to revert to the earlier version:

   SYSRQF
   SYSLOG
   SYSKPL
   SYSKEY
   SYSFSI

Instead of backing up the SYSLOG as a file, you can, for example, simply back up the content of SYSLOG in CSV format and then evaluate the log records at a later date. To do this, use the command SHOW-FT-LOGGING-RECORDS LAYOUT=*CSV.
Following installation, openFT then works with a new SYSLOG file.

3. Make sure that there are no requests in the request queue when you perform the version change.
   Reason: Requests are not taken over from V12.0 to V12.1. As a result, unfinished requests are lost on migration and may even cause requests for which their is no correspondence in the local system to persist on partner systems.
4. Delete the files SYSOPF, SYSRQF and SYSPTF, and possibly also SYSLOG (if the old log records are no longer required in the new version). The SYSLOG file can only be taken over to an offline log file.
5. Leave the SYSKPL and SYSKEY files so that they can be taken over into the new version, if authentication is still to be used there. In this case, you must **not** delete these files!

After completion of the IMON installation of openFT (and possibly of openFT-FTAM or openFT-FTP), openFT recreates the files SYSRQF, SYSOPF and SYSPTF on the first access attempt.

Taking over settings

If you want to continue to work with the old operating parameter settings, partner list entries and, if applicable, FTAC settings, you should run the procedures generated during the back-up operation (see point 1 on page 20).

Notes on openFT-FTP

When using openFT-FTP, you should note that after installation the FTP server is automatically deactivated.

If you want to use openFT-FTP for inbound requests, you have the following possibilities:

- You take over the operating parameter settings from openFT V12.0 provided that the openFTP server has been activated there.
- You activate the FTP server manually using the command MODIFY-FT-OPTIONS ...,ACTIVE-APPLICATIONS. When doing this, note the following:
  ACTIVE-APPLICATIONS=*ALL activates all protocols,
  ACTIVE-APPLICATIONS=*FTP only activates the FTP protocol.

For details, see manual "openFT (BS2000) - Command Interface ".

An error message FTR0852 will be output if the configuration files SYSOPF, SYSPTF, SYSRQF, SYSLOG, SYSFSA and SYSKEYdo not match with the current openFT version.

2.1.3 Installation of the command interface for POSIX

openFT (BS2000) supports the command interface which is provided in Unix and Windows systems also in POSIX. This means that you can use the openFT functions in BS2000 from within a POSIX system. With a very few exceptions (e.g. Unicode support of file names), the commands have the same function scope as they do in Unix or Windows systems.

For details see the manual "openFT (Unix and Windows systems) - Command Interface" and the manual "openFT (Unix and Windows systems) - Installation and Operation".

Installation

The library SINLIB.OPENFT.121 is required for installation. This contains the installation script and all the components needed for POSIX installation. The POSIX subsystem must be active for installation.

You install the command interface as follows:

– Call the START-POSIX-INSTALLATION command.

– Select “Install packages in POSIX” from the menu and enter the data in the “BS2000 POSIX package installation” screen which now opens. For details, see the POSIX manual “Basics for Users and System Administrators”.

Note

To support the command FTPING in POSIX the TCL package for BS2000 is provided along with the openFT package. Please note the following:

1. The TCL utility package is provided with the name SINLIB.OPENFT.121.TCL-UTIL. This package should be installed using the tool POSIX installation program, which is started by START-POSIX-INSTALLATION.
2. The openFT TCL utility package takes up 9 MBytes below the directory /opt/openFT. /opt/openFT is usually placed on the root file system.

2.2 Startup

The FT administrator's tasks have been simplified in openFT as of V10.0 since the request queue and partner list files are created when openFT is installed.

2.2.1 Preparing the FT system

ID and required PAM pages

For the first installation, an ID with the name SYSFJAM and the default catalog ID must be created for openFT on the home pubset of the processor. If you are running multiple openFT instances on your system, you must set up the configuration user IDs of the instances so that they are the same as SYSFJAM (the restriction concerning the home pubset does not apply here). The IDs should be set up in a manner that prevents a SET-LOGON-PARAMETERS command being entered. The number of PAM pages required by this ID depends on:

– the size of the request files and partner lists used and the required functionality, i.e., whether FTAC functionality is to be used. The openFT request queue SYSRQF has a default size of 12690 PAM pages while the option file SYSOPF requires 6 PAM pages and the partner list SYSPTF 1824 PAM pages. The FTAC file SYSFSA occupies at least 501 PAM pages.

– the size of the log file SYSLOG (at least 501 PAM pages) which, in turn, depends on the number of transfer requests handled and on which sets are engaged (i.e. only FT sets, only FTAC sets, or both).

– the size of the SYSFSI, SYSKPL, and SYSKEY files (by default, a total of 54 PAM pages), and on the number of key pairs (SYSPKF files) created.

– the type and number of trace functions activated.

It is therefore advisable to allow PAM page overruns for the ID SYSFJAM.

Access to public keys

In order to be able to access public keys, the FT administrator needs access to the SYSPKF files and the SYSKEY library on SYSFJAM or on the configuration user ID. If he/she does not have privileges granting him/her access to operating system resources, the FTAC admission profiles should be set up to grant him/her access.

Starting and stopping the FT subsystem

openFT requires a subsystem catalog entry containing a subsystem declaration with the load time set to “AT-CREATION-REQUEST”. The FT subsystem must be explicitly loaded in a startup procedure (e.g. CMDFILE).

When an FT instance is stopped, (particularly by using /STOP-SUBSYSTEM FT) all the file locks held by openFT (see page 51) are cleared and, on loading an instance (e.g. by using /START-SUBSYSTEM FT), the locks are reset for files affected by existing requests. The FT or system administrator must therefore observe the following:

– On starting the FT subsystem, all pubsets that contain data that is to be used in the event of a restart must be available.
  On the other hand, the loading must also occur early enough to ensure that the files to be transmitted are protected in time. This also applies to the transfer files of all configured openFT instances.
– Unloading an FT instance should be done as late as possible, but before the export of the pubsets on which the files to be transmitted are located.

Result lists

The job class JBCLLST should be generated with a small maximum processing time and, if possible, a high selection priority for printing result lists. This job class should be accessible to all FT users. The high priority (JOBPRIORITY operand in job class setup) ensures that jobs of this type are quickly started. A low maximum processing time (CPU-TIME operand in job class setup) prevents these jobs blocking the processor for a prolonged time.

Follow-up processing

For follow-up processing initiated by the openFT, you should generate the job class JBCLJOB with low maximum processing time and, if necessary, a high selection priority. If you do not do this the default job class will be used for follow-up processing. You should start extended, CPU-intensive follow-up jobs as enter jobs using the job classes which are available as standard in the BS2000-System.

Depending on the protocols configured, openFT can be reached via port 1100 (openFT protocol), 4800 (FTAM protocol), 21 (FTP protocol) and 11000 (FTADM protocol). To do this, openFT itself creates a BCMAP entry on START-FT. The following command is set for initializing mapping:

/BCMAP FUNCT=INIT,MAXMAP=500

If initialization is to be done using other values, it must take place before the first START-FT command.

2.2.2 Entering partners in the partner list

In openFT V10.0, the network description has been replaced by a partner list. The partner list is set up by openFT on installation. Following a new installation, it is empty.

Although the entry of partners in the partner list is optional, this offers significant advantages. These include simplified addressing for users, the central administration of partner addresses and enhanced security since you can assign individual properties such as security level or partner check level to partner systems, for example if authentication is required. Authentication requires partners to be entered in the partner list.

Consequently, you should enter partners with special characteristics in the partner list immediately after installation. The following options are available:

– If you are upgrading from an older openFT version, start the command procedure which you created with START-OPENFTPART or SHOW-FT-PARTNERS in the older openFT version. The previous entries are taken over into the partner list.

– ADD-FT-PARTNER command
  This enters a new partner in the partner list.

In the operating parameters, you can specify that only the named partners from the partner list may be addressed (corresponds to the state up to openFT V9.0).

For further details on administering partners during operation, see section “Administering partners” on page 42.

2.2.3 Starting and stopping openFT

Starting openFT

openFT is started with the START-FT command. Care must also be taken to ensure that all pubsets are available, as otherwise any locally submitted request that requires an unavailable pubset will be terminated with an error message. If this happens, the user cannot be informed of the circumstances by an event list or a job variable.

START-FT starts all applications that have been activated using the command MODIFY-FT-OPTIONS .. ACTIVE-APPLICATIONS=.

If multiple instances are used on one computer, each instance must be started individually. Individual instances can be set up so that they are automatically started on executing the command START-SUBSYSTEM, see the section “Using openFT instances” on page 74).

If the openFT option HOST-NAME is not set at start time then the real BCAM host is used. If multiple instances have to be started in a system then the host name must be set using the /MODIFY-FT-OPTIONS command at all but one of them.

Stopping openFT

Using STOP-FT terminates openFT in the current instance. When file transfer is terminated, non-restartable requests are aborted. Local requests continue to be accepted even after STOP-FT. The requests are stored in the request queue until openFT is restarted. When START-FT is entered again, the requests are processed in sequence.

2.3 Installing openFT-AC

The installation of openFT V12.1 is required for the installation of openFT-AC V12.1.

2.3.1 Initial installation

Delivery of openFT-AC takes place using the software delivery and information system SOLIS2. Installation takes place via IMON. If required, the installation contains BS2000-specific jobs such as MSGFILE update, subsystem catalog entries, and importing the SDF syntax file.

For the security of the SYSFSA file on the configuration user ID of the current openFT instance, it is recommended that you activate the class 2 ENCRYPTION option for password encryption. SYSFSA contains the settings for admission sets and admission profiles.

2.3.2 Version change

If an older version of openFT-AC is installed on your computer, it is recommended that you delete all product files of the old version with the exception of SYSFSA. Profiles and admission sets from the predecessor version V11 can be transferred unchanged. For all older versions, it is advisable to export the FTAC admission profiles and sets using the EXPORT-FTAC-ENVIRONMENT command.

The openFT-AC system file SYSFSA (on all instance IDs) can be moved from V12.0 to V12.1. If it should become necessary to migrate back to the previous version, you should first back this file up or export it to an FTAC export file, because once SYSFSA has been opened by openFT V12.1, backward migration is no longer possible.

2.4 Configuring openFT-AC

Authorization of the FTAC administrator

It is recommended that the position of administrator for openFT-AC be given to a user in the system who is responsible for data protection in a BS2000 system, since he will know where what protection measures are required .

The FTAC administrator function is assigned by means of the SECOS privilege FTAC-ADMINISTRATION. It may also be assigned to several user IDs at once.
For BS2000 installations without SECOS, the administration attribute has a fixed assignment to the user ID TSOS.

FTAC administrators who possess both the FTAC administration and TSOS privilege have the following additional rights:

– If they import profiles (for any user ID), they can select whether the profiles will be immediately available and unrestricted, or whether they will be locked.
– If they create profiles for external IDs then these are also immediately available. This means that they can create valid transfer admissions even if they do not know the LOGON password of the target ID. This method can be used to set up profiles that remain valid after the LOGON password is modified.
– They can therefore also modify the transfer admissions of existing profiles with external IDs without knowing the profile owner’s password.

Adapting the default admission set

After the installation of openFT-AC, 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 system. This is because as long as no other admission sets are made with MODIFY-FT-ADMISSION-SET, the default admission set is valid for all user IDs. The maximum security level 0 for the basic functions (inbound send, inbound receive, inbound follow-up processing, inbound file management, outbound send, outbound receive) means that these basic functions may not be used. The FTAC administrator must therefore use the command MODIFY-FT-ADMISSION-SET to raise the values of the default admission set.

Default security levels for partners

The FT administrator can use the MODIFY-FT-OPTIONS command (SECURITY-LEVEL operand) to define default security levels for all the partner systems entered in the partner list. The administrator can either enter a fixed value or specify *BY-PARTNER-ATTRIBUTES to indicate that the security level is set automatically: partners which are authenticated by openFT are assigned security level 10. Partners which are known in BCAM (i.e. they are addressed via their BCAM name) are assigned security level 90. All other partners are assigned security level 100.

This automatic assignment can also be activated on a partner-specific basis using the operands of the same name:
ADD-FT-PARTNER and MODIFY-FT-PARTNER...,SEC-LEV=*BY-PART-ATTR

This automatic assignment always applies to partners that are not in the partner list.

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:

   ```
   /MOD-FT-AD␣*STD,MAX-LEV=100
   ```

   More information on the command MODIFY-FT-ADMISSION-SET can be found in the manual "openFT (BS2000) - Command Interface ".

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

   ```
   /MODIFY-FT-ADMISSION-SET 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 openFT-AC is only effective for connected products such as openFT or FTP. If other file transfer products without an openFT-AC connection are also being used, a more comprehensive and coordinated security concept would be advisable.

3 Operating 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 (e.g. TRANSIT),
- 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 2 2. Increase TRANSPORT-UNIT-SIZE 3. Reduce CONNECTION-LIMIT
Computer and network overloaded	1. Set PROCESS-LIMIT to 2 2. Reduce CONNECTION-LIMIT
Throughput inadequate	1. Increase TRANSPORT-UNIT-SIZE
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 MODIFY-FT-OPTIONS. 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.

The setting PROCESS-LIMIT=*NONE corresponds to the setting PROCESS-LIMIT= CONNECTION-LIMIT. A separate task is generated for each connection.

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. TRANSPORT-UNIT-SIZE has no effect for links to FTAM partners. 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 CANCEL-FILE-TRANSFER command internally.

3.2 Administering code tables

As FT administrator, you must consult with the BS2000 system administrator to ensure that the required code tables are available on the system.

3.2.1 XHCS support by openFT

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

3.2.1.1 Binary file transfer

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

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

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

If the receiving system is not an openFT on a BS2000 system, the CCSN is ignored.

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

3.2.1.2 File transferred as text file

This section describes the scenario in which no CCS has been assigned to the text file in the transfer request.

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

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

Sending

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

Example

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

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

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

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

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

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

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

Example

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

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

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

Receiving

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

- The sending system is also an **openFT in a BS2000 system with XHCS**

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

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

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

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

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

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

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

Example

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

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

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

- The sending system is **not** an **openFT in a BS2000 system**

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

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

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

  Example

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

3.3 Administering requests

MODIFY-FILE-TRANSFER	Modify the order and priority of outbound requests within the request queue
CANCEL-FILE-TRANSFER	– Delete FT request from the request queue – Abort file transfer while in progress
SHOW-FILE-TRANSFER	Information on FT requests and their properties
MODIFY-FT-PARTNER	Activate or deactivate locally submitted requests for a particular remote system (STATE operand)

The FT administrator can use the CANCEL-FILE-TRANSFER ... FORCE-CANCELLATION command to force the full, unconditional cancellation of a request and its removal from the request file, if necessary without any negotiation with the partner system.

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:

ADD-FT-PARTNER	Add new partner system entries to the partner list
MODIFY-FT-PARTNER	Modify partner system entries in the partner list
REMOVE-FT-PARTNER	Remove partner systems from the partner list
SHOW-FT-PARTNERS	View information on partner systems in the partner list and back up the partner list (page 43)
START-OPENFTPART	Back up the partner list (page 43)
MODIFY-FT-OPTIONS	Define the global FTAC security level, enable/disable dynamic partners

For links with FTAM partners assumes that the transport system permits parallel connections. The remote systems are identified via their presentation addresses. Either BS2000 or the FTAM partner can initiate file transfer.

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:

```
ADD-FT-PARTNER 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 ADD-FT-PARTNER command) is a differently set attribute.

If you reset all the attributes for a partner of this type to the default values with MODIFY-FT-PARTNER 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 SHOW-FT-PARTNERS command or the START-OPENFTPART command:

- SHOW-FT-PARTNERS outputs the partner entries in the form of MODIFY-FT-PARTNER commands. To do this, specify the OUTPUT=*SYSLST(LAYOUT=*BS2-PROC) operand.

  The output can be redirected to a file by means of the ASSIGN-SYSLST command. To make the procedure executable, the first column of the output file must then be removed in an editor.

- START-OPENFTPART also outputs the partner entries to a file in the form of MODIFY-FT-PARTNER commands with the difference that the first column is already removed.

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

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:

MODIFY-FT-OPTIONS, IDENTIFICATION=instance-id

Instance identification of partners

Store instance IDs of partner systems in the partner list using the IDENTIFICATION parameter of the ADD-FT-PARTNER command, or MODIFY-FT-PARTNER. 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.

CREATE-FT-KEY-SET	creates an RSA key pair set for the local openFT instance
SHOW-FT-KEY	shows the attributes of all keys in the local system
UPDATE-FT-PUBLIC-KEYS	updates the public keys
DELETE-FT-KEY-SET	deletes local RSA key pair sets

MODIFY-FT-KEY	modifies RSA key attributes
IMPORT-FT-KEY	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 CREATE-FT-KEY-SET 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:

```
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 MODIFY-FT-OPTIONS 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 UPDATE-FT-PUBLIC-KEYS 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 UPDATE-FT-PUBLIC-KEYS.

If you want to replace a key pair set with a completely new one, you can create a new key pair set using CREATE-FT-KEY-SET. 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 DELETE-FT-KEY-SET.

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 SYSKEY library on the configuration user ID 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 IMPORT-FT-KEY 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 45. This key can be used in the same way as a key generated with CREATE-FT-KEY-SET 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 47.

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 BS2000 as type D PLAM elements in the **SYSKEY** library on the configuration user ID 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 IMPORT-FT-KEY 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 SHOW-FT-KEY ...SELECT=*PAR(PARTNER-NAME=...) to display the keys of partner systems and filter on expiration date.

Modifying the keys of partner systems

You can use the MODIFY-FT-KEY 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 44). 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
  MODIFY-FT-OPTIONS ... PARTNER-CHECK=*TRANSPORT-ADDRESS
- only for specific partners, using
  ADD-FT-PARTNER ... PARTNER-CHECK=*TRANSPORT-ADDRESS or
  MODIFY-FT-PARTNER ... PARTNER-CHECK=*TRANSPORT-ADDRESS

The global setting is valid for all partners with the value PARTNER-CHECK=*BY-FT-OPTIONS (default in the ADD-FT-PARTNER).

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 (MODIFY-FT-OPTIONS 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 MODIFY-FT-OPTIONS ...,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 (MODIFY-FT-OPTIONS command KEY-LENGTH parameter).

Using the CREATE-FT-KEY-SET 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 IMPORT-FT-KEY.

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.

In BS2000, if the CRYPT subsystem is installed and started in addition to openFT-CR, then openFT itself does not encrypt the file content, but allows CRYPT to handle the encryption. This considerably enhances performance.

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 MODIFY-FT-OPTIONS 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 and FTAM 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

Prior to version V8.0, FT products in BS2000 protected a file to be transferred only during the active transmission, i.e., when the file was opened by openFT using DVS. Consequently, if the transmission was interrupted or even if the transmission had not yet begun, both files involved could be potentially accessed and modified. Such changes could not always be detected on restarting openFT, thus resulting in the creation of inconsistent receive files.

As of V8.0, openFT uses an operating system mechanism to protect transfer files (however, this protection is not possible for library elements and Posix files):

– When a file transfer request is accepted, a lock is set on each file to be transferred as early as possible. Only read access is granted to other users for the send files; no access is permitted for the receive files.

– This lock remains set - so long as the FT subsystem is loaded - until the request has completed.

– The BS2000 command SHOW-FILE-LOCK indicates whether a file has been locked by openFT and, if it is, shows the transfer ID (or, when sending, possibly a list of transfer IDs) of the request involved. Such locks, and other file locks as well, can be reset by the system administrator at his/her own discretion in emergency situations by using the command REMOVE-FILE-ALLOCATION.

– Using SHOW-FILE-TRANSFER ... PUBSET=, the FT administrator can have all the requests displayed that have locked files on a defined pubset. The administrator can selectively delete these requests using CANCEL-FILE-TRANSFER ... PUBSET=.

On unloading an FT instance (STOP-SUBSYSTEM FT or DELETE-FT-INSTANCE), all the locks held by openFT are cleared and reset upon reload (START-SUBSYSTEM FT or CREATE-FT-INSTANCE) for all files affected by existing requests. For information on what the FT or system administrator must therefore take into consideration, see section “Starting and stopping openFT” on page 26.

In addition to this mechanism, openFT also implicitly checks the integrity of the transferred data by communicating with openFT partners version V8.1 and later. The scope is defined in the transfer request:

– In the case of requests with encryption, the transferred file content is also checked (TRANSFER-FILE ... DATA-ENCRYPTION = *YES).

– In the case of requests without encryption, an integrity check of the file content can be activated explicitly
(TRANSFER-FILE ... DATA-ENCRYPTION = *ONLY-DATA-INTEGRITY.

– If neither encryption nor the integrity check are activated then only the integrity of the request description data is checked
(TRANSFER-FILE ... DATA-ENCRYPTION = *NO).

If an error is detected then restartable requests attempt the transfer again. Requests that cannot restart are aborted.

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:

SHOW-FT-OPTIONS	Information on operating parameters
SHOW-FT-PARTNERS	Information on partner systems
SHOW-FT-LOGGING-RECORDS	Information on log entries
SHOW-FILE-TRANSFER	Information on file transfer status
SHOW-FT-INSTANCE	Information on openFT instances
SHOW-FT-MONITOR-VALUES	Show monitoring data from openFT operation

The SHOW-FT-OPTIONS command furnishes information on the current settings of the operating parameters.

SHOW-FT-PARTNERS 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 57.

The command SHOW-FT-LOGGING-RECORDS can be used to display the logs of file transfer requests.

SHOW-FILE-TRANSFER 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 SHOW-FT-INSTANCE, the FT administrator can find out which openFT instances exist in the system and have their characteristics and status displayed.

SHOW-FT-MONITOR-VALUES outputs the monitoring values from openFT operation. To do this, monitoring must be activated by means of MODIFY-FT-OPTIONS.

3.6.1 FT logging

The following 3 commands are available for the FT logging function:

DELETE-FT-LOGGING-RECORDS	– Deleting log records
	– Deleting offline log files
MODIFY-FT-OPTIONS	– 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.
SHOW-FT-LOGGING-RECORDS	– 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 SYSLOG.Lyymmdd.Lhhmmss on the configuration user ID of the openFT instance (default: $SYSFJAM).

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.

The SYSLOG files are created by the FT system with second allocation 500, its net size depends on the number of logging records it contains.

Changing the log file and administering offline log files

You can change the log file using the command MODIFY-FT-OPTIONS 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 DELETE-FT-LOGGING-RECORDS 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 SHOW-FT-LOGGING-RECORDS command to output all FT logging records of his/her system to SYSOUT or SYSLST. 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 MODIFY-FT-OPTIONS 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 MODIFY-FT-OPTIONS 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 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. These messages can be consulted on the manual server (*http://manuals.ts.fujitsu.com*) using an HTML application. 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 have the routing code '@', which means that they must be explicitly requested, for example, using the following command:

```
/MOD-MSG-SUBSCRIPTION ADD-MSG-ID=(FTR0301,FTR0307,FTR0340,FTR0341)
```

Messages for monitoring partner systems

FTR0301 Partner '(&00)' entered state NOCON

FTR0302 Partner '(&00)' entered state ACTIVE

FTR0303 Partner '(&00)' entered state LUNK

FTR0304 Partner '(&00)' entered state RUNK

FTR0305 Partner '(&00)' entered state INACTIVE

FTR0306 Partner '(&00)' entered state AINAC

FTR0307 Partner '(&00)' may be unreachable

FTR0308 Partner '(&00)' does not allow any more inbound requests

FTR0309 Partner '(&00)' added

FTR0310 Partner '(&00)' removed

FTR0311 Partner '(&00)') entered state LAUTH

FTR0312 Partner '(&00)' entered state RAUTH

FTR0313 Partner '(&00)' entered state DIERR

FTR0314 Partner '(&00)' entered state NOKEY

FTR0315 Partner '(&00)' entered state IDREJ

Messages for monitoring openFT

FTR0320 abnormal termination initiated

FTR0360 openFT control process started

FTR0361 openFT control process terminated

Messages for monitoring the request queue

FTR0330 Request queue 85 percent full

FTR0331 At least 20 percent of request queue unoccupied

Messages for monitoring requests

FTR0340 Transfer '(&00)' successfully completed

FTR0341 Transfer '(&00)' terminated with error

3.6.3 Monitoring openFT using a job variable

You can monitor an openFT instance by an automatically populated MONJV. The job variable is located under the configuration user ID of the instance concerned (e.g. $SYSFJAM) and has the name MONJV.OPENFT. The content of the job variable complies with the standard for MONJVs. The following information is provided by openFT:

Position 1-2 = state:

$R	open FT is active.
$T	openFT has been terminated normally.
$A	openFT has been terminated abnormally.

Position 5-8 = TSN of the control process of the instance involved.

The job variable is created the first time START-FT is issued and used thereafter. If it is not possible to modify the job variable for some reason, this does not have any effect on openFT operation. A diagnostics record is simply created in order to subsequently identify the cause.

3.6.4 SNMP management for openFT

SNMP stands for **S**imple **N**etwork **M**anagement **P**rotocol and was developed as the protocol for network management services in TCP/IP networks. openFT permits you to centrally monitor and administer one or more openFT systems from one central management station using graphical interfaces. A prerequisite for SNMP-based openFT management is the installation of the products SNMP Management ≥ V6.0, SNMP Basic Agent BS2000 V6.0 (SBA-BS2) and SNMP Standard Collection BS2000 V6.0 (SSC-BS2).

Detailed information can be found in the respective user manuals.

To support automatic monitoring, some events which are not direct responses to user input are reported by openFT via a console message. Console messages can also be used to generate SNMP traps for automatic FT monitoring using SNMP.

If the file transfer subagent is used then openFT itself can also generate SNMP traps (without having to use console messages).

The file transfer subagent is used to:
– start and stop openFT (BS2000)
– acquire system parameter information
– change the public key for encryption
– output statistics
– diagnostic control
– output partner information

The proprietary MIB for openFT contains objects for the management tasks listed above. The objects for starting and stopping, changing the public key for encryption, and for diagnostic control also provide write access.

3.6.4.1 Starting and stopping openFT

MIB definition/ object identifier	**Access**	**Meaning**
ftStartandStop 1.3.6.1.4.1.231.2.18.1.1.0	read-write	Start / Stop

openFT is started and stopped via the openFT subagents by setting the value “START” or “STOP” respectively. A read access returns information on the current FT system state.

3.6.4.2 System parameters

MIB definition/ object identifier	Access	Meaning
ftSysparVersion/ 1.3.6.1.4.1.231.2.18.2.1.0	read-only	Version
ftSysparTransportUnitSize/ 1.3.6.1.4.1.231.2.18.2.2.0	read-write	Transport Unit Size
ftSysparTaskLimit/ 1.3.6.1.4.1.231.2.18.2.3.0	read-write	Task Limit
ftSysparConnectionLimit/ 1.3.6.1.4.1.231.2.18.2.4.0	read-write	Connection Limit: maximum number of transport connections that can be reserved for the execution of FT requests
ftSysparPartnerCheck/ 1.3.6.1.4.1.231.2.18.2.6.0	read-write	Partner Check
ftSysparMaxInboundReqs/ 1.3.6.1.4.1.231.2.18.2.12.0	read-write	Max Inbound Requests: maximum number of inbound requests per partner system
ftSysparMaxLifeTime/ 1.3.6.1.4.1.231.2.18.2.13.0	read-write	Max Request Lifetime: maximum lifetime (in days) in the request queue

Further information of the output values can be found in the description of the SHOW-FT-OPTIONS command.

3.6.4.3 Public key for encryption

MIB definition/ object identifier	Access	Meaning
ftEncryptKey/ 1.3.6.1.4.1.231.2.18.3.1.0	write-only	An entry of “create-new-key” or “1” causes a new public key to be generated.

3.6.4.4 Statistics

MIB definition/ object identifier	Access	Meaning
ftStatSuspend/ 1.3.6.1.4.1.231.2.18.4.1.0	read-only	Requests in a SUSPEND state
ftStatLocked/ 1.3.6.1.4.1.231.2.18.4.2.0	read-only	Requests in a LOCKED state
ftStatWait/ 1.3.6.1.4.1.231.2.18.4.3.0	read-only	Requests in a WAIT state
ftStatActive/ 1.3.6.1.4.1.231.2.18.4.4.0	read-only	Requests in an ACTIVE state
ftStatCanceled/ 1.3.6.1.4.1.231.2.18.4.5.0	read-only	Requests in a CANCELD state
ftStatFinished/ 1.3.6.1.4.1.231.2.18.4.6.0	read-only	Requests in a FINISHED state
ftStatHold/ 1.3.6.1.4.1.231.2.18.4.7.0	read-only	Requests in a HOLD state
ftStatLocalReqs/ 1.3.6.1.4.1.231.2.18.4.8.0	read-only	Async requests in the local system
ftStatRemoteReqs/ 1.3.6.1.4.1.231.2.18.4.9.0	read-only	Requests in the remote system

A description of the output values can be found in the description of the SHOW-FILE-TRANSFER command.

3.6.4.5 Diagnostic control

MIB definition/ object identifier	Access	Meaning
ftDiagStatus/ 1.3.6.1.4.1.231.2.18.5.1.0	read-write	on / off
ftDiagFtamPartners/ 1.3.6.1.4.1.231.2.18.5.2.0	read-write	on / off
ftDiagOpenftPartners/ 1.3.6.1.4.1.231.2.18.5.3.0	read-write	on / off
ftDiagFtpPartners/ 1.3.6.1.4.1.231.2.18.5.4.0	read-write	on / off
ftDiagSynRequests/ 1.3.6.1.4.1.231.2.18.5.5.0	read-write	on / off
ftDiagAsynRequests/ 1.3.6.1.4.1.231.2.18.5.6.0	read-write	on / off
ftDiagLocRequests/ 1.3.6.1.4.1.231.2.18.5.7.0	read-write	on / off
ftDiagRemRequests/ 1.3.6.1.4.1.231.2.18.5.8.0	read-write	on / off
ftDiagOptionsNobulk/ 1.3.6.1.4.1.231.2.18.5.9.0	read-write	on / off

Please also read the description of the MODIFY-FT-OPTIONS command.

3.6.4.6 Partner Information

MIB definition/ object identifier	Access	Meaning
ftPartnerName/ 1.3.6.1.4.1.231.2.18.8.1.1.1.0	read-only	Name of the FT partner
ftPartnerType/ 1.3.6.1.4.1.231.2.18.8.1.1.2.0	read-only	FT protocol used by the partner
ftPartnerState/ 1.3.6.1.4.1.231.2.18.8.1.1.3.0	read-write	Status of the FT partner: act (1), inact (2), nocon (3), lunk (4), runk (5), adeact (6), ainact (7) lauth (8) rauth (9) dierr (10) nokey (11) idrej (12)
ftPartnerAddress/ 1.3.6.1.4.1.231.2.18.8.1.1.10.0	read-only	Address of the partner system

Only a status update for one partner is supported at present, and only the values act, inact and adeact may be specified.

3.6.4.7 Traps

Object name/ object identifier	Trap No.	Explanation
Enterprise = sniFTTraps		
ftStopTrap/ 1.3.6.1.4.1.231.2.18.6.0.1.0	1	TRAP is sent if openFT is terminated
ftPartnerStateTrap/ 1.3.6.1.4.1.231.2.18.6.0.4.0	4	TRAP is sent if the partner status has changed
ftPartnerUnreachableTrap/ 1.3.6.1.4.1.231.2.18.6.0.5.0	5	May not be possible to access partner
ftStartTrap/ 1.3.6.1.4.1.231.2.18.6.0.6.0	6	TRAP is sent after start of openFT
ftRequestQueueUpperLimitTrap/ 1.3.6.1.4.1.231.2.18.6.0.7.0	7	TRAP is sent if the FT request queue is more than 85% full
ftRequestQueueLowerLimitTrap/ 1.3.6.1.4.1.231.2.18.6.0.8.0	8	TRAP is sent if at least 20% of the FT request queue is free again
ftRequestSuccessfulTrap/ 1.3.6.1.4.1.231.2.18.6.0.9.0	9	TRAP is sent if an FT request is sent successfully
ftRequestErrorTrap/ 1.3.6.1.4.1.231.2.18.6.0.10.0	10	TRAP is sent if an FT request is terminated with an error
ftSubsystemStartTrap/ 1.3.6.1.4.1.231.2.18.6.0.11.0	11	TRAP is sent if the FT subsystem has been started
ftSubsystemStopTrap/ 1.3.6.1.4.1.231.2.18.6.0.12.0	12	TRAP is sent if the FT subsystem has been stopped

3.6.4.8 Trap groups and trap controls

The traps of the openFT subagent can be gathered together into groups that are represented by the following MIB objects. This means that you can enable or disable the sending of traps for the individual trap groups as follows (trap send status “on” or “off”):

– Specification 2 (“on”): the traps for the group in question are sent.
– Specification 1 (“off”): the traps for the group in question are not sent.

MIB definition/ object identifier	Access	Affected traps
ftTrapsSubsystemState/ 1.3.6.1.4.1.231.2.18.10.1.0	read-write	– ftSubsystemStartTrap – ftSubsystemStopTrap
ftTrapsFTState/ 1.3.6.1.4.1.231.2.18.10.2.0	read-write	– ftStartTrap – ftStopTrap

MIB definition/ object identifier	Access	Affected traps
ftTrapsPartState 1.3.6.1.4.1.231.2.18.10.3.0	read-write	– ftPartnerStateTrap
ftTrapsPartnerUnreachable/ 1.3.6.1.4.1.231.2.18.10.4.0	read-write	– ftPartnerUnreachableTrap
ftTrapsRequestQueueState/ 1.3.6.1.4.1.231.2.18.10.5.0	read-write	– ftRequestQueueUpperLimitTrap – ftRequestQueueLowerLimitTrap
ftTrapsTransSucc/ 1.3.6.1.4.1.231.2.18.10.6.0	read-write	– ftRequestSuccessfulTrap
ftTrapsTransFail/ 1.3.6.1.4.1.231.2.18.10.7.0	read-write	– ftRequestErrorTrap

3.6.4.9 Trap information

The MIB of the openFT subagent contains definitions of MIB objects which are sent together with the traps.

MIB definition/ object identifier	Access	Explanation
ftRequestID/ 1.3.6.1.4.1.231.2.18.9.1.0	not-accessible	Transfer ID of the request
ftRequestInitiator/ 1.3.6.1.4.1.231.2.18.9.2.0	not-accessible	Initiator of the request: local (1), remote (2)
ftRequestPartnerName/ 1.3.6.1.4.1.231.2.18.9.3.0	not-accessible	Partner
ftRequestUserID/ 1.3.6.1.4.1.231.2.18.9.4.0	not-accessible	User ID of submitter
ftRequestFileName/ 1.3.6.1.4.1.231.2.18.9.5.0	not-accessible	Name of the file for transfer
ftRequestError/ 1.3.6.1.4.1.231.2.18.9.6.0	not-accessible	Error in request

3.6.5 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.

3.6.5.1 Configuring monitoring

You configure monitoring using the MODIFY-FT-OPTIONS 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 SHOW-FT-OPTIONS. The MONITOR row indicates whether monitoring is activated and shows any criteria used for selection.

3.6.5.2 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 SHOW-FT-MONITOR-VALUES to show monitoring data locally.

SHOW-FT-MONITOR-VALUES 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 SHOW-FT-MONITOR-VALUES 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 SHOW-FT-MONITOR-VALUES 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:

```
/CREATE-FT-PROFILE 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.

3.7.1 Administrating admission sets

For the administration of admission sets, openFT-AC offers the FTAC administrator the following commands:

MODIFY-FT-ADMISSION-SET	Modify admission sets
SHOW-FT-ADMISSION-SET	Show admission sets

As the FTAC administrator, you are responsible for the following tasks:

- You define the standard admission set using the command MODIFY-FT-ADMISSION-SET 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 SHOW-FT-ADMISSION-SET 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 MODIFY-FT-ADMISSION-SET.

3.7.2 Administrating admission profiles

For the administration of admission profiles, openFT-AC offers the FTAC administrator the following commands:

CREATE-FT-PROFILE	create admission profile
DELETE-FT-PROFILE	delete admission profile
MODIFY-FT-PROFILE	modify admission profile
SHOW-FT-PROFILE	show admission profile

The FTAC administrator has the option of modifying foreign admission profiles:

– He can view them with the command SHOW-FT-PROFILE. 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 DELETE-FT-PROFILE. 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 MODIFY-FT-PROFILE, or conversely revoke privileges.

– He can also modify them with MODIFY-FT-PROFILE. If the FTAC neither possesses the TSOS 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 MODIFY-FT-PROFILE <profile> TRANSFER-ADMISSION=*OLD-ADMISSION(VALID=*YES).

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 CREATE-FT-PROFILE.
2. The FTAC administrator views the admission profile with the command SHOW-FT-PROFILE to determine if the profile presents a threat to data security.

   Example

   ```
   /SHOW-FT-PROFILE NAME=PROFPROD,
                    SELECT-PARAMETER=(OWNER-IDENTIFICATION=STEVEN),
                    INFORMATION=*ALL
   ```

   Short form:

   ```
   /SHOW-FT-PROF PROFPROD,SEL=(,STEVEN),INF=*ALL
   ```

   The output has the following form:

   ```
   %PROFPROD
   % IGN-MAX-LEV = (IBR)
   % FILE-NAME   = PROFIT
   % USER-ADM    = (STEVEN,M4711DON,OWN)
   % PROC-ADM    = SAME
   % FT-FUNCTION = (TRANSFER-FILE, MODIFY-FILE-ATTRIBUTES,
                     READ-FILE-DIRECTORY)
   % LAST-MODIF  = 2017-01-15 08:24:49
   ```

   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 CREATE-FT-PROFILE 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

   ```
   /MODIFY-FT-PROFILE NAME=PROFPROD,
                      SELECT-PARAMETER=(OWNER-IDENTIFICATION=STEVEN),
                      PRIVILEGED=*YES
   ```

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:

EXPORT-FTAC-ENVIRONMENT	output FTAC environment to file
IMPORT-FTAC-ENVIRONMENT	transfer FTAC environment from file
SHOW-FTAC-ENVIRONMENT	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.

  Any existing privileges must be explicitly set up again on the new computer, and the admission profiles must be explicitly released if you (as the FTAC administrator) do not possess the TSOS privilege.

  On the other hand, if you have the TSOS privilege, you can specify on importing whether the profiles will be imported with unmodified attributes or not.

- You can display the contents of a backup file with the command SHOW-FTAC-ENVIRONMENT.

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.

```
/EXPORT-FTAC-ENVIRONMENT 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.

```
/SHOW-FTAC-ENVIRONMENT 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):

```
/SHOW-FTAC-ENVIRONMENT 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:

```
/IMPORT-FTAC-ENVIRONMENT FROM-FILE=STEVEN.FTAC.BKUP
```

Since Sylvester does not possess the TSOS privilege, he must now privilege Steven's profile:

```
/MOD-FT-PRO PROFPROD,,(,STEVEN),PRIV=*Y
```

Finally, Steven must release the imported profiles before he can work with them. This would not be necessary if Sylvester possessed the TSOS privilege.

```
/MODIFY-FT-PROFILE 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 54.

The display of FTAC logging records can not be turned off. However, the MODIFY-FT-OPTIONS 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 SHOW-FT-LOGGING-RECORDS 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 SHOW-FT-LOGGING-RECORDS command, see manual "openFT (BS2000) - 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 to carry over the functionality of the openFT to another computer, which is already running openFT.

After installing openFT, the **standard instance** exists on each computer. This instance is atypical in that it cannot be deleted by instance management commands. Its application data is located on the default pubset under ID $SYSFJAM. When instances are displayed (SHOW-FT-INSTANCE), the standard instance is always displayed first.

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 therefore requires a configuration user ID with the characteristics that are described for the SYSFJAM ID (see section “Preparing the FT system” on page 23).
  Exception: It is not necessary that the configuration user ID is located on the home pubset.
- Each instance requires its own network address; this always remains the same, independent of the real host.
  The host name must therefore be stored in the options using the MODIFY-FT-OPTIONS command. This (virtual) BCAM host must always be accessible under the same network address. In order to prevent the BCAM connection setup from automatically being passed to the real host instance when an instance fails to start, BCAM aliasing should be disabled for the $FJAM and $FTAM applications.

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, reps, 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 Commands for administrating openFT instances

As the openFT administrator, you can create, modify, and delete instances. In addition, you can set and get information on instances (like a user). The creation, modification and deletion of instances is only possible via the SDF interface, not via the POSIX command interface.

- Creating an instance

  Using the CREATE-FT-INSTANCE command, you can create an instance.

  If an instance is created, an entry is made in the administration file. This entry consists of the name of the instance and the pubset and user ID in which the files required for operation are stored (the request file, partner list, etc.). All the initialization tasks are carried out in the same way as during START-SUBSYSTEM. In the event you have also specified the AUTOMATIC-START option, then openFT is subsequently and immediately started in this instance.

- Modifying an instance

  With the MODIFY-FT-INSTANCE command, you can rename an instance and modify its AUTOMATIC-START characteristics.

- Deleting an instance

  With the DELETE-FT-INSTANCE command, you can delete an instance. Deleting an instance removes the administration entry for the instance. All the variable data (the request file, partner list, etc.) of this instance continue to exist and can be re-activated by repeating the CREATE-FT-INSTANCE command. Any attempt to access a deleted instance is denied with FTR0236.

  The standard instance cannot be deleted.

- Setting an instance

  Using the SET-FT-INSTANCE command, you can select the openFT instance with which you would like to work (see the user guide). This setting is then valid for all the SDF commands set under this task or program interface calls and remains valid until the task is ended or until the next SET-FT-INSTANCE command. If you want to continue to work with the set instance in a Posix shell then it is necessary to call the following command after starting the shell:

  . ftseti

  The dot (.) followed be a blank is mandatory!

  It is therefore advisable to record this command in the */etc/profile* file.
  If no SET-FT-INSTANCE command is given in a task, then work proceeds using the standard instance.

- Displaying instance information

  Using the SHOW-FT-INSTANCE command, you can request information regarding the instances, see the user guide.

- Set or display the BCAM host

  Using the MODIFY-FT-OPTIONS ...,HOST-NAME command, you can assign the current instance a BCAM host. This BCAM host will be used for communication of openFT. By doing this, an instance allows itself to be assigned a fixed transport address, which is independent of the computer on which the instance is running.

  On executing the SHOW-FT-OPTIONS command, the name of the BCAM host with which the instance is working is displayed.

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 (STOP-FT).

– Unload the instance on the original computer (DELETE-FT-INSTANCE). This unlocks all of the files required by openFT (request file, transfer files, etc.).

– Import the variable files, the network address (virtual BCAM host) 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).

  It is recommended to import all files of the configuration user ID when changing over.

– Load the instance on the destination computer (CREATE-FT-INSTANCE).

– Start the instance on the destination computer (if this does not occur automatically, then use SET-FT-INSTANCE, START-FT).

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 pubsets that are accessed by requests must be available. All requests whose pubsets 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, from the point of view of the BCAM, the same host names for partner computers must be available and they must refer to the same partner computer. The network address of the (virtual) 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 name of the instance must be the same on all of the computers, since, for example, it is used for qualifying temporary files.

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:

  ```
  /ASSIGN-SYSLST TO=OPTION-FILE
  /SH-FT-OPT OUTPUT=*SYSLST(*BS2-PROC)
  /ASSIGN-SYSLST TO=*PRIMARY
  ```

  The first column of the file created (in the example, this is OPTION-FILE) contains print control characters. This means that you must subsequently delete the first column.

- Backing up partner list entries:

  ```
  /START-OPENFTPART PARTBS2.SAV
  ```

- Backing up the FTAC environment:

  ```
  /EXPORT-FTAC-ENV FTACBS2.SAV
  ```

3.10 FTAM characteristics on BS2000-Systems

3.10.1 Mapping FTAM attributes to the real file system

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

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

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

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

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

3.10.1.1 Inbound mapping of FTAM attributes

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

Attribute group	FTAM attributes	Mapping in BS2000 (inbound receive)	Modify FTAM attributes
Kernel group	permitted actions READ-FILE INSERT-DATA-UNIT REPLACE-FILE EXTEND-FILE ERASE-DATA-UNIT READ-ATTRIBUTES CHANGE-ATTRIBUTES DELETE-FILE	FTAM catalog	permitted locally 1)
	universal class number GRAPHIC GENERAL IA5 VISIBLE string significance UNCHANGED VARIABLE FIXED NOT-SIGNIFICANT maximum string length	FTAM catalog FTAM catalog FTAM catalog	permitted locally 1) permitted locally 1) permitted locally 1)
	document type FTAM1 FTAM3	FTAM catalog	permitted locally 1)
Storage group	file availability IMMEDIATE DEFERRED future file size storage account	FTAM catalog ignored ignored	inbound permitted for DMS files, not permitted for POSIX files and lib members not permitted not permitted

Attribute group	FTAM attributes	Mapping in BS2000 (inbound receive)		Modify FTAM attributes
Security group	ActionList (of 1ACE)	with (BASIC)ACL	without (BASIC)ACL	
	READ-FILE	(BASIC-) ACL=r	ACCESS=READ	inbnd. permitted [5]
	INSERT-DATA-UNIT	not permitted	not permitted	not permitted
	REPLACE-FILE	w	WRITE	inbnd. permitted [5]
	EXTEND-FILE	w	WRITE	inbnd. permitted [5]
	ERASE-DATA-UNIT	w	WRITE	inbnd. permitted [5]
	READ-ATTRIBUTES	[3]	READ	not permitted [2]
	CHANGE-ATTRIBUTES	[4]	READ	not permitted [2]
	DELETE-FILE	w	WRITE	inbnd. permitted [5]
	LEGAL-QUALIFICATION	ignored		not permitted

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

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

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

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

[5] ACL attributes cannot be modified.

The following file attributes are derived from the current BS2000 file attributes:
- file name
- file size
- future filesize
- identity of creator (always identical to owner ID)
- date and time of last read access (corresponds to LADATE and LATIME)
- date and time of last attribute modification (corresponds to CRDATE/CRTIME)
- date and time of last modification (corresponds to CRDATE/CRTIME)
- access control
- string significance

Other attributes are only partially supported by openFT (BS2000). As the responder, openFT (BS2000) does not return any value for the following file attributes (*no value available*):
- identity of last modifier
- identity of last reader
- identity of last attribute modifier
- storage account
- legal qualification

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

3.10.1.2 Inbound mapping the document type

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Access protection

openFT supports the security group of the virtual filestore. This provides an effective protection mechanism against unauthorized access to files.

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

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

3.10.1.3 Outbound mapping of the document type

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

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

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

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

Entries in the TRANSFER-FILE(NCOPY) command:

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

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

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

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

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

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

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

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

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

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

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

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

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

Entries in the TRANSFER-FILE(NCOPY) command:

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

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

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

FTAM-3 files are not converted.

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

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

Entries in the TRANSFER-FILE(NCOPY) command:

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

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

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

FTAM-3 files are not converted.

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

Entries in the TRANSFER-FILE(NCOPY) command:

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

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

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

Case c2:
Transfer a file to the FTAM partner,

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

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

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

Case c3:
Transfer a file to the FTAM partner,

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

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

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

Case c4:
Transfer a file to the FTAM partner,

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

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

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

FTAM-3 files are not converted.

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

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

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

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

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

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

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

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

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

Case e:
Transfer a text file from the FTAM partner

Entries in the TRANSFER-FILE(NCOPY) command:

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

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

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

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

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

Case f1:
Transfer a binary file from the FTAM partner

Entries in the TRANSFER-FILE(NCOPY) command:

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

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

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

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

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

FTAM-3 files are not converted.

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

Entries in the TRANSFER-FILE(NCOPY) command:

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

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

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

Entries in the TRANSFER-FILE(NCOPY) command:

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

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

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

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

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

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

Case h:
Transfer a library member from the FTAM partner

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

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

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

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

4 Remote administration

4.1 Configuring an openFT instance on the BS2000 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 BS2000 system using the REMOTE-ADMINISTRATION function:

```
/CREATE-FT-PROFILE 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 BS2000 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:

```
/EXECUTE-REMOTE-FTADM-CMD 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 EXECUTE-REMOTE-FTADM-CMD in the manual "openFT (BS2000) - 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:

```
/EXECUTE-REMOTE-FTADM-CMD 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 98.

command
: Specifies the administration command to be executed on the openFT instance. For further details, see command EXECUTE-REMOTE-FTADM-CMD in the manual "openFT (BS2000) - 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 BS2000 system using the SHOW-FT-LOGGING-RECORDS command and delete them with the DELETE-FT-LOGGING-RECORDS command (provided that you have the appropriate permission to do so). Please refer to the manual "openFT (BS2000) - 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 MODIFY-FT-OPTIONS 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 BS2000 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 MODIFY-FT-OPTIONS command:

```
/MODIFY-FT-OPTIONS ...                                          -
/        ,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 (START-FT / STOP-FT)
- Change of partner status
- Unavailability of partners
- Change of request management status
- Successfully completed requests
- Failed requests

To do this, use the MODIFY-FT-OPTIONS 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 BS2000 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 98.

If you wish to view the most recent 10 ADM traps, enter the following remote administration command:

```
/EXECUTE-REMOTE-FTADM-CMD 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 information:

- Creating diagnostic records
- Trace function
- Recovering from hung FT and FTAC subsystems

In POSIX environment, you can also use the ftping command to test the response from a remote partner. The interface of ftping on BS2000 systems is exactly similar to the ftping syntax on Unix and Windows platforms. The syntax is displayed when you enter ftping without options.

5.1 Creating diagnostic records

If, despite due care and attention, an error occurs that neither the FT administrator nor the BS2000 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 section “Trace function” on page 105
- if applicable the result list of the request that triggered the error
- CONSLOG file of the entire session (also from partner system if possible)
- general information as for BS2000 system error on openFT or the BS2000 operating system, DCAM, PLAM, SDF and, if required, openFT-FTAM, openFT-AC for BS2000, NFS and POSIX:
  1. system version number,
  2. loader - subversion number / code,
  3. list of all rep corrections used;
- version of the FT partner and details of the transport system (e.g. DCAM, CCP / CMX, VTAM, etc.)
- system dumps requested under the TSN FTxx or FT server tasks
- system dumps after interrupts in the modules of the FT and FTAC subsystems

The SHOW-FT-DIAG command can be used to output any diagnostic codes written when the error occurred (together with time and date). In this case, SHOW-FT-DIAG supplies the following output:

```
/SH-FT-DIAG
% DATE           TIME      SSID  COMPONENT    LOCATION-ID        INFO
% 20170121       143307    FT    79/yfasdia   3/EuisyMsg         fd00000c
```

SHOW-FT-DIAG INF=*ALL allows additional information on the current state of openFT to be obtained. This only makes sense, however, if it is done shortly after the problem arises.

5.2 Trace function

5.2.1 Controlling the trace function

The FT administrator uses the following commands to control the trace function:

ADD-FT-PARTNER	Add a remote system to the partner list
MODIFY-FT-OPTIONS	Modify operating parameters
MODIFY-FT-PARTNER	Modify partner properties in the partner list

The FT administrator uses the following commands to get information on the current settings:

SHOW-FT-OPTIONS	Information about operating parameters
SHOW-FT-PARTNERS	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 MODIFY-FT-OPTIONS command. You can differentiate by partner type (openFT, FTP, FTAM), 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 MODIFY-FT-PARTNER (or set before with ADD-FT-PARTNER).

The following table illustrates four typical cases of trace use.

MODIFY-FT-OPTIONS	ADD-FT-PARTNER/ MODIFY-FT-PARTNER	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 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.
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 ADD-FT-PARTNER is BY-FT-OPTIONS. The global settings are thus taken over from MODIFY-FT-OPTIONS.

The following table indicates the interrelations between the most important MODIFY-FT-OPTIONS and ADD-/MODIFY-FT-PARTNER trace settings.

MODIFY-FT-OPTIONS	ADD-FT-PARTNER/ MODIFY-FT-PARTNER	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

MODIFY-FT-OPTIONS	ADD-FT-PARTNER/ MODIFY-FT-PARTNER	Effect
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

5.2.2 Evaluating traces

openFT generates trace files for the configuration user ID of the openFT instance (default: $SYSFJAM).

Format of the trace files

The file names end with the suffix .FTTF and have the following format:

– Smddhhmm.Sssccc.I000.FTTF'
  Control task.

– Smddhhmm.Sssccc.Iiii.FTTF'
  Server task for inbound and asynchronous outbound requests, i= 001,002, ...

– Ymddhhmm.Sssccc.Pnnnn.FTTF'
  User task for synchronous outbound requests.

mddhhmm.Sssccc 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). nnnn is the TSN of a task for outbound requests.

The trace files contain openFT, FTAM, FTP and ADM requests that have been processed in the corresponding task.

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.:
  S8101010.S33222.I001.FTTF (first trace file)
  S8101010.S33222.I001.L001.FTTF (continuation file)

START-FTTRACE

Traces are evaluated with START-FTTRACE:

START-FTTRACE
INPUT = <filename 1..54>
,**OUTPUT** = <filename 1..54> / *SYSLST
,TRACE-OPTION = <c-string 1..50 with-lower-case>
,SHOW-FILE = ***NO** / ***YES**
,PRINT-FILE = ***NO** / ***YES**

Operand description

INPUT = <filename 1..54>
Filename of the trace file to be evaluated $SYSFJAM.SYSFLF.D*yymmdd*.T*hhmmss.tsn.*

OUTPUT = <filename 1..54>
Filename of the output file.

OUTPUT = *SYSLST
Output to SYSLST, e.g. during preprocessing. This also implicitly sets the SHOW-FILE operand to *NO.

TRACE-OPTION = <c-string 1..50 with-lower-case>
Specifies the options for the trace evaluation in the following format:.

[-d] [-sl=n | sl=l | sl=m | sl=h] [-cxid=<context-id>] [-f=hh:mm:ss] [-t=hh:mm:ss]

-d
Specifies that the trace files are to be output in hexadecimal format (dump format).

CAUTION!
Data that is critical for security (transfer admissions, passwords etc.) is not “masked” in dump format. The specification of a security level or levels is irrelevant here.

-sl=n | **-sl=l** | **-sl=m** | **-sl=h**
Specifies the security level for the output:

n (no) No security requirements, i.e. all the data is output. This includes IDs, passwords, transfer admissions, 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.
This parameter is not relevant in the case of dump format.

-cxid=<context id>
Selects the trace entries on the basis of the context ID. If you omit *-cxid* or specify *-cxid=* without a context ID then all the trace entries are output.

-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 beginning 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.

SHOW-FILE =
Specifies if the evaluated trace file should be displayed with the command SHOW-FILE.

SHOW-FILE = *NO
The evaluated trace file is not displayed. Default value in batch mode.

SHOW-FILE = *YES
The evaluated trace file is displayed. Default value in dialog mode.

PRINT-FILE =
Specifies whether the evaluated trace file should be printed.

PRINT-FILE = *NO
The evaluated trace file is not printed.

PRINT-FILE = *YES
The evaluated trace file is printed.

5.3 Recovering from hung FT and FTAC subsystems

In some cases, e.g. when system errors occur, it may not be possible to unload the subsystems. This may be due to the following reasons:

– The subsystem is in a LOCKED state, since the associated holder task can no longer be used, e.g. after a system dump.

– The subsystem cannot be unloaded because some tasks are still connected. This typically occurs when FT cannot be exited (the tasks with the TSNs FTC* and the FT server tasks (job name FTSP) do not disappear) when connected tasks enter permanent wait states, or when tasks are “permanently pending” after system dumps due to insufficient disk space.

In such exceptional cases, the system administrator can resort to some special resources to unload the subsystems and thus save the BS2000 session. These resources are described in the DSSM manual. The following points discuss some of the aspects to be observed when unloading the FT subsystems.

WARNING!

There is always a certain risk involved in using any such resources. There is essentially no way of guaranteeing that all error states are fully described here. The ultimate responsibility always lies with the system administrator!

A subsystem in a LOCKED state can be removed from the system by using the command /UNLOCK-SUBSYSTEM. Note, however, that this does **not** call the subsystem-specific uninstallation routine and therefore has the following consequences for the FT subsystem:

- No file locks held by FT are released, so all user files locked by FT will remain locked. These locks can be removed explicitly by the system administrator with REMOVE-FILE-ALLOCATION-LOCKS or will disappear implicitly at the next BS2000 startup.

A subsystem to which tasks are still connected can be unloaded with /STOP-SUBSYSTEM FORCE=YES if required, provided the attribute FORCED=ALLOWED is first assigned with the command /MOD-SUBSYSTEM-PARAMETERS. By default, FT subsystems do not have this attribute.

This approach causes any restarted tasks that are still connected to run onto a system dump. Such system dumps are of no use whatsoever for any diagnostic purposes and may hence be discarded. FT tasks which are still connected and which blocked in a waiting state will run within at most 10 minutes onto the system dump.

The subsystem should never be restarted as long as there are existing tasks which were still connected at the time of forcing the subsystems to unload!

6 Appendix

6.1 Accounting records

Structure of the FT accounting records

The FT accounting record is divided into four parts:

1. record definition
2. identification section
3. basic information
4. variable information

The record sections contain the displacement, length and format of the data field.

The **field number** identifies the sequence number within the part of the record written.

The **displacement** is the position of the data field relative to the beginning of the part of the record that has been written.

The **length** is the length of the data field in bytes.

The **format** is the format of the data field:

A = alphanumeric (including $, #, and @)

B = binary number

C = printable characters

F = file name for BS2000

Z = unpacked decimal number (0...9)

– = undefined

1. Record definition section

The record definition section contains the record identifier, the time of day, the length of the identification section and the length of the basic information.

Field No.	Displ.	Length	Format	Meaning
1	00	4	A	Record identifier ‘FTR0‘
2	04	8	- /B	Time stamp of the time-of-day clock
3	0C	2	B	Length of the identification section
4	0E	2	B	Length of the basic information
5	10	4	-	Reserved

Layout of the record definition section

2. Identification section

The identification section contains the user ID, account number and job number (TSN).

Field No.	Displ.	Length	Format	Meaning
1	00	8	A	User ID
2	08	8	A	Account number
3	10	4	Z	Job number (TSN) (This field applies only to locally issued requests.)

Layout of the identification section

3. Basic information

The basic information includes

– Date and time when the FT request was stored,

– Date and time when the transfer ended,

– Result of the transfer,

– Details of the start of follow-up processing,

– Name of the remote system,

– Indication as to whether the accounting record was written by the local or the remote system,

– Identification of the FT request,

– Number of disk accesses,

– Number of bytes written to or read from disk,

– Number of bytes sent to or read from the accounting record.

Field No.	Displ.	Length	Format	Meaning
1	00	12	Z	Time when the file transfer request was stored (format: yymmddhhmmss; this field applies only to locally issued requests)
2	0C	12	Z	Time when the transfer ended (format: yymmddhhmmss)
3	18	1	C	Result of the transfer: + : successful execution - : execution with errors 0 : not used
4	19	1	C	Result of the start of follow-up: + : successful execution - : execution with errors 0 : not used
5	1A	8	A	Partner name
6	22	1	A	Specifies whether the request was issued in local or remote system: L : the request was submitted in the local system R : the request was submitted in the remote system
7	23	11	Z	Transfer ID
8	2E	2	-	Reserved
9	30	4	-	Reserved

Layout of the basic information

Field No.	Displ.	Length	Format	Meaning
10	34	4	B	Number of disk accesses
11	38	8	B	Number of bytes on disk
12	40	8	B	Number of bytes in network

Layout of the basic information

4. Variable information

Field No.	Displ.	Length	Format	Meaning
1	00	2	B	Number of extensions = 4
2	02	2	B	Displacement of the record extension for the file name from the start of record
3	04	2	B	Displacement of the record extension for the library member name from the start of record
4	06	2	B	Displacement of the record extension for the century part of the time specification from the start of the record
5	08	2	B	Displacement of the record extension for the CPU time from the start of the record
The variable information includes the file name and the name of the library member.				

Header of the variable section

Field No.	Displ.	Length	Format	Meaning
1	00	2	A	Extension identification = ‘FN‘
2	02	1	B	Extension type = x‘00‘
3	03	1	B	Length of the file name
4	04	see field 3	F	File name
If a displacement is set to 0, the corresponding record extension has not been specified.				

Record extension for the file name

Field No.	Displ.	Length	Format	Meaning
1	00	2	A	Extension identification = ‘MN‘
2	02	1	B	Extension type = x‘00‘
3	03	1	B	Length of extension (not including identification, type and length field)
4	04	8	A	Library member type
5	0C	24	A	Library member version
6	24	8	Z	Library member variant
7	2C	1	B	Length of library member name
8	2D	see field 7	A	Library member name

Record extension for the library member name

Field No.	Displ.	Length	Format	Meaning
1	00	2	A	Extension identification = ‘YY‘
2	02	1	B	Extension type = x‘00‘
3	03	1	B	Length of extension (not including identification, type, length field) = 4
4	04	2	Z	Time at which the request was stored in the form yy (see field 1 in the basic information)
5	06	2	Z	Time of transfer in the form yy (see field 2 in basic information)

Record extension for the century part of the time specification

Field No.	Displ.	Length	Format	Meaning
1	00	2	A	Extension identification = ‘MS‘
2	02	1	B	Extension type = x‘00‘
3	03	1	B	Length of extension (not including identification, type, length field) = 4
4	04	4	B	Number of machine commands required in the local system by this request (in units of 10,000 commands)

Record extension for CPU time

Index