

Fujitsu Software

openFT (Unix- and Windows-Systeme)

Installation and Operation

System Administrator Guide

Valid for:
openFT V12.1C40

November 2022

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: bs2000services@ts.fujitsu.com.

Certified documentation according to DIN EN ISO 9001:2015

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 9001:2015.

Copyright and Trademarks

Copyright © 2022 Fujitsu Technology Solutions GmbH.

All rights reserved.

Delivery subject to availability; right of technical modifications reserved.

All hardware and software names used are trademarks of their respective manufacturers.

Table of Contents

Installation and Operation (Unix and Windows Systems)	8
1 Preface	9
1.1 Brief description of the product	10
1.2 Target group	11
1.3 Concept of openFT manuals	12
1.4 Changes since the last version	14
1.4.1 Changes for all platforms	15
1.4.2 Changes for Unix and Windows platforms	18
1.4.3 Changes for Unix platforms	19
1.4.4 Changes for BS2000 systems and z/OS	20
1.4.5 Changes for z/OS	21
1.4.6 New functions that are only available in the openFT Explorer	22
1.5 Notational conventions	23
1.6 Internet	24
2 Installation on Unix systems	25
2.1 Installation of openFT	26
2.1.1 New installation	28
2.1.2 Update installation of openFT	30
2.1.3 Installation of a patch	32
2.1.4 Installation in an alternative root directory (Solaris)	33
2.1.5 Automatic installation	35
2.1.6 Support of Openshift	36
2.1.7 Support of systemd	39
2.1.7.1 Single-user mode	40
2.1.7.2 Enabling ftalarm command	41
2.1.7.3 Modifying umask value	42
2.2 Important activities after installation	43
2.2.1 Checking the default settings	44
2.2.2 Importing configuration data	46
2.2.3 Disabling the automatic startup of openFT	47
2.2.4 Enabling the ftalarm command	48
2.2.5 Solaris SMF	49
2.2.6 Installing or uninstalling openFT-FTAM under HP-UX, AIX and Linux	52
2.2.7 Installing or uninstalling openFT-FTP under HP-UX, AIX and Linux	53
2.2.8 Authentication via PAM	54
2.2.9 Creating the partner list from TNS	56
3 Installation on Windows systems	57

3.1 Installation of openFT	58
3.1.1 New installation	60
3.1.2 Update installation of openFT	62
3.1.3 Installation of a patch	64
3.1.4 Unattended installation	65
3.1.5 Installation of the SNMP subagent	68
3.1.6 Changing and repairing openFT	69
3.1.7 Deinstallation of openFT, openFT-CR and CMX	70
3.2 Important activities after installation	71
3.2.1 Checking the default settings	72
3.2.2 Importing configuration data	74
3.2.3 Activating and disabling the automatic startup of openFT	75
3.2.4 Creating the partner list from TNS	76
4 Operating	77
4.1 Optimizing the operating parameters	79
4.2 Administering code tables	80
4.3 Starting and stopping the asynchronous openFT server	83
4.4 Setting operating modes (Windows systems)	84
4.4.1 Running the service under system rights	85
4.4.2 Running the service under user rights	86
4.5 Setting the protection bit for newly created files (Unix systems)	87
4.6 File access rights for newly created files (Windows systems)	88
4.7 File access under user rights (Unix systems)	89
4.8 Single-user mode (Unix systems)	90
4.8.1 Switching to single-user mode	91
4.8.2 Returning to multi-user mode	93
4.8.3 Note on the ftalarm function in single-user mode	94
4.9 Switching the language interface	95
4.9.1 Switching the language interface on Unix systems	96
4.9.2 Switching the language interface on Windows systems	97
4.10 Administering requests	98
4.11 Administering partners	99
4.11.1 Exporting the partner list	101
4.12 Security in FT operation	102
4.12.1 Authentication	103
4.12.1.1 Unique Instance Identifications	104
4.12.1.2 Creating and administering local RSA key pairs	105
4.12.1.3 Importing keys	107
4.12.1.4 Exporting public keys	108
4.12.1.5 Administering the keys of partner systems	109
4.12.1.6 Distributing the keys to partner systems	110

4.12.2	Extended authentication check	111
4.12.3	Encryption on data transfer	112
4.13	Monitoring and controlling FT operation	113
4.13.1	openFT logging	114
4.13.2	SNMP interface of openFT (Windows)	116
4.13.2.1	Activities after installation	117
4.13.2.2	Starting the openFT subagent on Windows	118
4.13.2.3	SNMP management for openFT (Windows)	119
4.13.3	Monitoring with openFT	123
4.13.3.1	Configuring monitoring	124
4.13.3.2	Displaying monitoring data	125
4.14	Administering the FTAC environment	128
4.14.1	Administering admission sets	129
4.14.2	Administering admission profiles	131
4.14.3	Saving and migrating the FTAC environment	132
4.15	openFT instances and cluster operation	133
4.16	Save and restore configuration data	135
4.17	Characteristics of FTAM on Unix and Windows systems	136
4.17.1	Mapping file access rights	137
4.17.1.1	Outbound requests	138
4.17.1.2	Inbound requests	139
4.17.2	Mapping FTAM attributes to the real file system	142
4.17.2.1	Inbound mapping of FTAM attributes	143
4.17.2.2	Inbound mapping the document type	146
4.17.2.3	Access protection	148
4.17.2.4	Outbound mapping of the document type	149
5	Central administration	151
5.1	Remote administration	152
5.1.1	Configuring the remote administration server	153
5.1.1.1	Defining the ADM administrator	154
5.1.1.2	Declaring an openFT instance as a remote administration server	155
5.1.1.3	Setting up admission profiles for accessing the remote administration server	156
5.1.1.4	Entering the openFT instances to be administered in the partner list	157
5.1.1.5	Creating a configuration file using the Configuration Editor	158
5.1.1.6	Creating a configuration file using a text or XML editor	161
5.1.1.7	Importing the configuration	172
5.1.1.8	Exporting and modifying a configuration	173
5.1.2	Configuring an openFT instance to be administered	174
5.1.2.1	Configuring an admission profile for an openFT instance as of V11.0	175
5.1.2.2	Configuring an admission profile for an openFT instance < V11.0	176

5.1.3 Issuing remote administration requests	177
5.1.3.1 Remote administration using the command interface	178
5.1.3.2 Remote administration using the openFT Explorer	180
5.1.4 Logging remote administration	183
5.2 ADM traps	184
5.2.1 Configuring the ADM trap server	185
5.2.2 Configuring ADM traps in the openFT instance	186
5.2.3 Viewing ADM traps	187
5.3 Example of an XML configuration file	189
6 Troubleshooting and Diagnosis	194
6.1 What if	195
6.2 Creating diagnostic records	201
6.3 Trace function	202
6.3.1 Trace files	203
6.3.1.1 Activating/deactivating trace functions	204
6.3.1.2 Viewing trace files	206
6.3.1.3 Evaluating trace files with ftrace	207
6.4 Additional diagnostic information	209
6.4.1 Displaying diagnostic records	210
6.4.2 Message file for console commands	211
6.4.3 Output diagnosis information with diaginfo (Windows systems)	212
6.5 Code tables	213
6.5.1 Code table EBCDIC.DF.04	214
6.5.2 Code table ISO 8859-1	215
7 Appendix	216
7.1 Important CMX commands (Unix systems)	217
7.1.1 tnsxcom - Create the TS directory	218
7.1.2 tnsxprop - Output properties of TS applications	219
7.2 Entering transport system applications in the TNS	221
7.2.1 TNS entries created automatically	223
7.2.2 Definition of the local TS application for openFT-FTAM	225
7.2.3 Definition of a remote TS application for openFT	226
7.2.3.1 Sample entries for openFT partners (Unix systems)	227
7.2.3.2 Sample entries for openFT partners (Windows systems)	228
7.2.4 Definition of remote TS applications for openFT-FTAM	230
7.2.4.1 Sample entries for FTAM partners (Unix systems)	232
7.2.4.2 Sample entries for FTAM partners (Windows systems)	233
7.3 openFT in a cluster with Unix based systems	235
7.3.1 Example 1: one fail-safe instance	236
7.3.2 Example 2: Fail-safe capability for both computers in the cluster	240
7.3.3 Notes for using TNS	243

7.4 The openFT instance concept in a Windows cluster	244
7.4.1 Sample	245
7.4.1.1 Installation of openFT	246
7.4.1.2 Configuration of resource-specific openFT properties of Cluster	247
7.4.1.3 Configuration of openFT	248
7.4.1.4 Operations with the individual openFT Instance	250
7.4.1.5 Use of the Windows cluster as an openFT Server	251
7.4.2 Configuring resource-specific openFT properties	252
7.5 FarSync X.25 transport system under Linux and Windows systems	260
7.5.1 Direct support of the FarSync X.25 for Windows systems	261
7.5.2 Support of the FarSync X.25 for Linux systems	262
7.5.3 Configuring the FarSync X.25 transport system in openFT	263
7.6 Sample files	264

Installation and Operation (Unix and Windows Systems)

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_64)
- Microsoft® Windows™ 10, Windows Server 2016 and 2019
- z/OS (IBM®)

1.1 Brief description of the product

The openFT product range comprises the following products:

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

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

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

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

The manual applies to the file transfer products openFT(Unix systems) and openFT(Windows).

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

openFT allows the use of the following transport protocols:

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

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

- **FTAC:**

FTAC provides extended system and data access protection. FTAC stands for File Transfer Access Control. On BS2000 systems, FTAC is provided by the add-on product openFT-AC. On z/OS, FTAC is provided by the add-on product openFT-AC. On BS2000 systems and on z/OS, FTAC is provided by the add-on product openFT-AC. On Unix and Windows systems, FTAC is integrated in openFT.
- **openFT-FTAM (not available on z/OS):**

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

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

openFT-CR is required up to version V12.1B if encrypted file transfer is requested.

! Attention

As of openFT version V12.1C, openFT-CR is no longer delivered because the functionality has been integrated in openFT V12.1C.

All references to openFT-CR therefore only apply to openFT versions <= V12.1B.

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 Unix or Windows system.

In addition, the manual is intended for the remote administrator who wants to start up and to run the openFT system as a remote administration server.

The manual covers Linux systems and Oracle Solaris systems as well as porting to other Unix platforms such as AIX or HP-UX.

i As of openFT version V12.1B, the release is only for Linux x86_64. Since openFT version V12.1C20, Solaris (Sparc) is also included in the release again. Further platforms are available on request from your Fujitsu sales representative.

The operating system-dependent differences are described in detail in the Release Notices supplied on the internet and the respective product CD.

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

openFT (BS2000) - Installation and Operation

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

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

openFT (z/OS) - Installation and Operation

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

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

openFT (Unix and Windows Systems) - Command Interface

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

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

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

openFT (BS2000) - Command Interface

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

- the conventions for file transfer to computers with different operating systems

-
- the openFT commands on BS2000 systems
 - the messages of the various components

openFT (z/OS) - Command Interface

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

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

openFT (BS2000) - Program Interface

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

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

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

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

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

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

1.4 Changes since the last version

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

- i** 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.

- BS2000 and z/OS systems (from openFT V12.1C): the new option FILE-NAME-ENCODING supports different character encoding for the specification of remote file names, pre, post and follow-up processing for transfer commands for the following commands:

- TRANSFER-FILE-SYNCHRONOUS / FTSCOPY
- TRANSFER-FILE / FTACOPY
- CREATE-REMOTE-DIR / FTCREDIR
- DELETE-REMOTE-DIR / FTDELDIR
- MODIFY-REMOTE-DIR-ATTRIBUTES / FTMODDIR
- MODIFY-REMOTE-FILE-ATTRIBUTES / FTMOD
- SHOW-REMOTE-FILE-ATTRIBUTES / FTSHW
- DELETE-REMOTE-FILE / FTDEL
- EXECUTE-REMOTE-CMD / FTEXEC
- EXECUTE-REMOTE-FTADM-CMD / FTADM.

- 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.

- Unicode for C and JAVA

- BS2000 and z/OS systems: for library elements the log records display the element name, type and version in addition to the library name .

-
- 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-FT-PARTNER/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.

- Encryption for file management requests

For file management requests the file and directory list attributes can be transferred encrypted..This property can also set in admission profiles.

To permit this, the following commands have been modified:

Unix and Windows systems:

- *ftshw*: New option *-c*
- *ftcrep* and *ftmodp*: New option *-cm*
- *ftshwp*: New parameter FILE-AT-ENC in the long output and new parameter *FileAtEnc* in the csv output.

BS2000 and z/OS systems:

- SHOW-REMOTE-FILE -ATTRIBUTES and FTSHW:
New operand FILE-ATTR-ENCRYPTION
- CREATE-FT-PROFILE and MODIFY-FT-PROFILE as well as FTCREPF and FTMODPRF: New operand FILE-ATTR-ENCRYPTION
- SHOW-FT-PROFILE and FTSHWPRF: New parameter FILE-AT-ENC in the long output and new parameter *FileAtEnc* in the csv output.

For getting multiple files form the remote system using *ft_mget*, GET-REMOTE-FILES or FTMGET the encryption option (*-c* or DATE-ENCRPTION) also applies to file and directory list attributes.

- From openFT V12.1C the CRYPT functionality is integrated in openFT. Therefore openFT-CR is no longer delivered.

Changes in openFT version V12.1C10:

- For local outbound requests, the long output and CSV output of log records display the remote filename in the new field REMOTE-FN. On BS2000 systems, the remote filename is also displayed in the OPS variable F-REMOTE-NAME.

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.
- The maximal length of the command string in the *ftexec* command has been raised to 8191.
- 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.

Changes in openFT version V12.1C20:

- Support of RSA keys with length 3072 and 4096.

Changes in openFT version V12.1C30:

- On Windows systems the openFT administrator rights can be transferred to any user. The *ftmodo* command has been extended for this purpose.

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.

- SNMP is no longer supported on Unix platforms.
- Support of *systemd*.

Changes in openFT version V12.1C10:

- Support of Openshift.

Changes in openFT version V12.1C30:

- Installation using license keys.

Changes in openFT version V12.1C40:

- On Unix systems, as already in Windows systems, the openFT administrator rights can be transferred to any user.
The *ftmodo* command has been extended for this purpose.

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.

Changes in openFT version V12.1C30:

- The PARM member of the z/OS parameter file has been changed as follows:
 - New key word TZSTRING for support of timezone handling to avoid missing lines when reading logging records that were written during the change from daylight saving time to winter time.

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.

italics

In running text, names, variables and values are indicated by italic letters, e.g. file names, instance names, menus, commands and command options.

i indicates notes.

! Indicates warnings.

1.6 Internet

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 on Unix systems

This chapter describes the installation and configuration of openFT on Unix systems.

i openFT is shipped with a communications manager until version V12.1B. In the following, this communications manager is always referred to as CMX (Communications Manager for Unix systems) even if different package names are used for the various platforms (such as CMX, PCMX, CMX.all, SMAWcmx, SMAWpcmx).

All those references are only valid for openFT versions \leq V12.1B.

2.1 Installation of openFT

The installation of openFT is performed under the login name *root*.

The installation technique of openFT depends on the operating system and is described in the respective Release Notice. There are three different types of installation depending on whether an FT version is already installed or which FT version is already installed on your computer:

- New installation
Your computer does not yet have an openFT or has an openFT < V11.0 on it.
- Update installation
Your computer has openFT version V11.0 or V12.0 installed.
- Installation of a correction version (patch)
Your computer has openFT version 12.1 installed.

What you need to observe before installing openFT ...

- Operation without CMX is supported as of openFT V12. If you want to work with CMX and CMX is not yet installed on the system then you must
 - install the CMX version present on the data medium and then
 - activate operation with CMX in openFT, e.g. using the `ftmodo -cmx=y` command.
- The language used by openFT (German, English) is set in accordance with the *LANG* environment variable in the case of a new installation (exception: English is always set in HP-UX systems). For more information, see [section "Switching the languageinterface"](#).
- If you want to encrypt file contents, you must also install openFT-CR V12.1 for Unix systems. This software is offered without a license at a fixed price. If an openFT-CR version < V11.0 is already installed, then you must first uninstall this version before installing openFT. You may only install openFT-CR V12.1 after openFT V12.1 has been installed.
- If you want to use the openFT-Script interface or the Java API then the J2SE™ Runtime Environment 7.0 (JRE 7.0 with update 25) or higher must be installed on your system.

The binary directory containing the *java* executable should be present under one of the following paths:

```
/opt/*/bin
/opt/**/bin
/usr/*/bin
/usr/**/bin or
/etc/alternatives/bin
```

The openFT installation procedure then creates the reference to the Java executable which is required in Unix systems in the openFT directory.

In other cases, the installation procedure issues a warning informing you that Java could not be found. It is recommended to install Java in one of the above-named directories and create the link to it. To do this, enter the following command:

```
ftsetjava @s
```

The `ftsetjava` command also allows you to check whether Java is installed and, if so, in which variant (`ftsetjava @a`) or check which Java variant is used (`ftsetjava` without parameters). In addition, you can set a path that is not located below the abovementioned paths (`ftsetjava file_name`).

- Instance directory

The instance directory is set up during installation and contains subdirectories for application-specific data for the corresponding openFT instance, such as the log file, key pair sets and trace files. By default, the default path name for the instance directory is `/var/openFT/instance` on Unix systems.

instance is the name of the corresponding instance. The standard instance named *std* always exists.

i When you create a new instance using `ftcrei`, you can select any path name for the instance directory.

The following sections describe which steps must be performed for the three installation variants by you as the system administrator as well as those which are handled automatically by the installation procedure.

2.1.1 New installation

If you have not yet installed any version of openFT on your computer or if a version < V11.0 is installed, the installation is a new installation.

Tasks required of the system administrator

1. If openFT version 10.0 and possibly add-on products are already installed, then you should proceed as follows:
 - Save admission profiles and admission sets that are still needed in an external file using *ftexpe*.
 - Save the partner list entries using *ftshwptn* in an external file and - if necessary - the openFT operating parameter settings using *ftshwo*.
 - Uninstall openFT-CR, openFT and the add-on products.
2. Install the openFT V12.1 product software.

When doing this, please note the following:

As of openFT version V12.1C30, openFT license keys (one for the openFT protocol and optionally one each for the FTAM and FTP protocol) can be entered at any time after installation using the openFT Explorer (Administration - License management) or the *ftaddlic* command.

If you do not enter a license key then openFT runs with full functionality as a demo version for 30 days. This demo version may only be used for evaluation purposes!
3. Import the saved configuration data as follows, if necessary:
 - Import the saved admission sets and admission profiles using *ftimpe*. If the admission sets and admission profiles were exported from an openFT version < V8.1 then all security levels in the admission sets that were previously set at 1 are automatically converted to 90. The standard admission set is re-set.
 - Import the saved partner list entries and operating parameter settings (if applicable) by executing the save file in the command prompt.

After these steps, openFT will be fully operational and will be activated at each system startup.

¹ As of openFT version V12.1B10, the release is only for Linux x86_64. Since openFT version V12.1C20, Solaris (Sparc) is also included in the release again. Further platforms are available on request from your Fujitsu sales representative.

Steps performed automatically

During installation, the following steps are carried out automatically:

- If CMX is installed then default TNS entries are generated for openFT if no TNS entries yet exist, otherwise they are adapted (see [section "TNS entries created automatically"](#)).

If CMX is subsequently installed then you can also use a tool to create default TNS entries at some later date, see [section "Entering transport system applications in the TNS"](#).
- The instance directory for the standard instance is set up, see [section "Installation of openFT"](#).

In this case, the operating parameters (e.g. maximum number of requests that can be processed simultaneously, maximum block length, scope of FT and FTAC logging, setting of the CCS, port numbers for the asynchronous inbound servers) are set to default values, see also [section "Checking the default settings"](#).

CMX operation, FTP server and the use of the TNS are deactivated.
- The name of the processor is entered as the processor name (corresponds to the output in *uname -n*).
- The DNS name of the computer (if one exists) is pre-set as the instance ID for the standard instance. When there is no DNS name, the name of the computer is used for the instance ID.
- A standard admission set permitting all file transfer functions is created.

-
- A key pair set is created (see [section “Creating and administering local RSA key pairs”](#)).
 - For openFT versions <= V12.1B (as of version V12.1C systemd is used):
The following startup and shutdown files are set up on the Linux, HP-UX and AIX platforms:
 - The startup and shutdown file that applies to all instances
(e.g. `/sbin/init.d/openFT` auf HP-UX)
 - The startup and shutdown file for the `std` instance
(path: `/var/openFT/std/etcinit/openFTinst`).

With the help of this file openFT is started automatically each time the system is started, and is terminated automatically each time the system is shut down (see also [section “openFT instances and cluster operation”](#)).

i On the Solaris platform, SMF is supported as of openFT V12.0, see [section “Solaris SMF”](#). As a result, no further startup or shutdown files are created.

- The man pages are installed as follows:
 - On the Solaris, AIX and HP platforms, the openFT man pages are installed in the same language as openFT as indicated by the LANG variable.
 - On Linux systems, the openFT German and English man pages are installed, i.e. users see the man pages in the language set for their login sessions (dependent on the LANG variable).
- The file transfer is started (but not on HP systems).
- The system searches for a suitable Java executable and this is notified to openFT. If no such system is found then you proceed as described in [section “Installation of openFT”](#).

2.1.2 Update installation of openFT

If openFT V11.0 or V12.0 is already installed, an update installation is performed.

Points to observe preparatory to an update installation

During an update installation, the following actions are carried out for all active instances including the standard instance:

- The log file is deleted. Therefore you should evaluate the log records before performing the update installation.
- Any running openFT-Script requests are aborted during installation. All old, aborted openFT-Script requests are not regarded as being restartable in the new openFT version. You should therefore complete all running openFT-Script requests before carrying out an update installation.
- Existing requests are deleted from the request queue unconditionally. If any follow-up processing was specified with the option *-lf=* in the submitted request, this is completed in the process.
- Existing trace files, if any, diagnostics records and console commands are deleted.

If you wish to continue using openFT instances that have been deactivated using *ftdeli*, you should activate them before the update installation using *ftcrei*. The corresponding instance file trees are then automatically updated during installation. If you do not do this, you must update these instances after installation using the *ftupdi* command (see the manual "openFT (Unix and Windows systems) - Command Interface").

Tasks required of the system administrator

1. Install openFT from the data medium.
2. On a system in which the openFT installation takes place in a dialog, you need to answer questions during installation asking you if you have a valid openFT-FTAM license and a valid openFT-FTP license. Only activate this option if you have a valid license for openFT-FTAM or openFT-FTP! Depending on the answers openFT-FTAM and/or openFT-FTP may or may not be installed.

These questions are not asked on HP, AIX and Linux systems. If you want to use the FTAM or FTP functionality on these systems, then you must activate openFT-FTAM and/or openFT-FTP via the *install.ftam* or *install.ftp* command after installing openFT. These commands are available in directory */opt/openft/bin/ftbin*, see also the manual "openFT (Unix and Windows systems) - Command Interface".
3. As of openFT version V12.1C30, openFT license keys (one for the openFT protocol and optionally one each for the FTAM and FTP protocol) can be entered at any time after installation using the openFT Explorer (Administration - License management) or the *ftaddlic* command.
If you do not enter a license key then openFT runs with full functionality as a demo version for 30 days. This demo version may only be used for evaluation purposes!
4. If you have made modifications in the old startup and shutdown files, then you must also
 - make them in the new startup and shutdown files in the case of an update installation under Linux, HP-UX or AIX
 - make them in SMF in the case of Solaris, see [section "Solaris SMF"](#).See also [section "openFT instances and cluster operation"](#).

Steps performed automatically

The following steps are performed automatically for an update installation:

- Running openFT processes and the openFT Explorer are terminated.

-
- openFT-Script requests are cancelled.
 - In the case of an openFT V10.0 update installation, the default TNS entries for openFT are handled as follows:
 - Default TNS entries from older openFT versions < V10.0 that are no longer required are deleted
 - Missing required default TNS entries are created.
 - Existing required default TNS entries remain unchanged.
 - The language setting from the previous version is used. On Linux platforms, however, the openFT man pages are installed in both German and English, i.e. users see the man pages in the language set for their login session.
 - The instance directories of currently existing instances including the standard instance are updated, i.e.:
 - The log file is deleted.
 - The old, instance-specific startup and shutdown files are backed up under */var/openFT/instance/etc/init/openFTinst.old* (*instance* = name of the instance). The new instance-specific startup and shutdown files are then read in on Linux, HP-UX and AIX platforms. On the Solaris platform, SMF is supported, see [section “SolarisSMF”](#). As a result, no further startup or shutdown files are created.
 - During this, the following configuration data are used:
 - Operating parameters (the operation CMX with remains activated)
 - Instance identification
 - partner list entries
 - The FTAM catalog
 - Admission sets and profiles
 - Key pair sets
 - Configuration data for central administration
 - openFT is started for those instances, for which it was started before the installation (not applicable on HP systems).
 - The system searches for a suitable Java executable and this is notified to openFT. If no such system is found then you proceed as described in [section “Installation of openFT”](#).

2.1.3 Installation of a patch

Installation of a patch means that openFT V12.1 is already installed on your computer. Please note the following:

- Any running openFT-Script requests are aborted during installation. You should therefore complete all running openFT-Script requests before installing a correction version.
- Any trace files, diagnostic records or files with console commands that may be present are deleted.

Tasks required of the system administrator

1. Install openFT V12.1 from the data medium.
2. On a system in which the openFT installation takes place in a dialog, you need to answer questions during installation asking you if you have a valid openFT-FTAM license and a valid openFT-FTP license. Only activate this option if you have a valid license for openFT-FTAM or openFT-FTP! Depending on the answers openFT-FTAM and/or openFT-FTP may or may not be installed.

This question is not asked on HP, AIX and Linux systems. openFT-FTAM and openFT-FTP are automatically installed on these systems if they were installed before.

3. As of openFT version V12.1C30, openFT license keys (one for the openFT protocol and optionally one each for the FTAM and FTP protocol) can be entered at any time after installation using the openFT Explorer (Administration - License management) or the `ftaddlic` command.

If you do not enter a license key then openFT runs with full functionality as a demo version for 30 days. This demo version may only be used for evaluation purposes!

Steps performed automatically

The following steps are performed automatically on installing a patch:

- Running openFT processes and the openFT Explorer are terminated.
- Running openFT-Script requests are cancelled. Interrupted `ftscript` runs from start up again when openFT is started.
- The admission profiles and admission sets, the log files, the startup and shutdown files (Linux, HP-UX, AIX) or the SMF connection (Solaris), the FTAM catalog, operating parameters and requests, the partner list, the key pair sets, and the configuration data for the central administration are taken over without changes for all openFT instances.
- The language setting from the previous version is used.
- The configuration data for the central administration is used.
- If you work on an HP, AIX or Linux system, then openFT-FTAM and openFT-FTP are automatically installed on these systems if they were installed in the previous version.
- openFT is started for those instances, for which it was started before the installation (not applicable on HP systems).

2.1.4 Installation in an alternative root directory (Solaris)

On the Solaris platform, openFT permits installation in an alternative root directory. This means that the files and directories of the openFT package are not installed in the root directory of the system that is currently running but in another directory that already contains an operating system environment and from which the system will subsequently be booted.

Installation in an alternative root directory is essential if support for live upgrade procedures is required. In the case of live upgrade procedures, the root file system is duplicated to an alternative root file system. The software (operating system update and additional software packages) is then installed in the alternative root file system from which the system is subsequently booted.

Variable openFT files

The variable openFT files are installed in the directory `/var/openFT`. It is not possible to work with a `/var` directory that is shared between the root directory and the alternative root directory.

The administrator is responsible for synchronizing the variable openFT files between the root file system and the alternative root file system, i.e. the administrator must synchronize the variable openFT files before starting the new system.

Installation of openFT

In the case of an update installation, the alternative root directory already contains an openFT version V11.0 or V12.0, or, if a correction version is to be installed, an openFT Version V12.1.

As of openFT version v12.1C30 during installation, you are asked to enter the license keys:

- If you specify a valid basic license key then the basic functionality of openFT will be enabled (openFT protocol, FTADM protocol and FTAC functionality). The optional license keys for the FTAM and FTP protocols are then queried. If you enter a valid license key for an optional protocol then this protocol is also enabled. You can skip this dialog box if you do not possess a valid key for FTP or FTAM.
If you enter a valid license key for a special version (e.g. CL or FTAM-CL version) then you will not be asked to enter a license key for FTAM and/or FTP because the supported function scope is already present in the special version.
- If you do not enter a license key then openFT runs with full functionality as a demo version for 30 days. This demo version may only be used for evaluation purposes!

Following installation, you can enter openFT license keys at any time using the openFT Explorer (*Administration - License Administration*) or the `ftaddlic` command.

Proceed as follows:

1. Install the openFT V12.1 product software in the alternative root directory.

For this purpose, change to the mounted directory and start the following procedure:

```
sh install.ft -r=<alternative root directory>
```

Example:

To install openFT in the directory `/altrroot` invoke the following command:

```
sh install.ft -r=/altrroot
```

When you do this, the fixed files and directories in the openFT package are installed in the alternative root directory, e.g. `/altrroot/opt/openFT`.

-
2. The following steps are necessary following a new installation or an update installation in order to generate the variable openFT files (new installation) or convert them to openFT V12.1 format (update installation):
 - a. Boot from the alternative root directory without starting openFT.

The automatic start-up of openFT via SMF is not yet activated.
 - b. Call the shell procedure *ftconfig*:

```
/opt/openFT/bin/ftbin/ftconfig
```

openFT is now fully installed.
 - c. Start openFT.

Following a correction installation, openFT is automatically configured and started the first time the new system is started up. In this case, it is not necessary for the administrator to call the shell procedure *ftconfig*, or start openFT.

Restrictions applying to an update installation

After an update installation, the following restrictions apply:

If you switch back to the original root file system then it is not possible to synchronize the variable openFT files because configuration files updated with openFT V12.1 cannot be converted back to an earlier version. This means that openFT requests and settings that have been made in the alternative root file system as well as new log records, trace files, diagnostics records etc. will be lost.

2.1.5 Automatic installation

i As of openFT version V12.1C30, prompts asking if FTP and FTAM should be installed are not displayed during installation on Solaris. Therefore, “FTAM” and “FTP” variables are not required inside of the response file, and if they are added they will have no effect.

On Solaris systems, you may also select automatic installation when installing openFT on some systems. In this case, installation is carried out without user prompts on screen. The additional data required for installation of openFT-FTAM and openFT-FTP are taken from the *response* file. A default response file with the following contents is integrated in the installation package:

```
FTAM='NO'
```

```
FTP='NO'
```

Meaning of the environment variable

FTAM specifies whether or not you are authorized to use the FTAM functionality, i.e. whether or not you have an openFT-FTAM license. In the standard response file, this variable is preset to *NO*, i.e. openFT-FTAM is not installed.

Other possible values:

YES, i.e. an openFT-FTAM license exists, the use of openFT-FTAM is activated.

FTP specifies whether or not you are authorized to use the FTP functionality, i.e. whether or not you have an openFT-FTP license. In the standard response file, this variable is preset to *NO*, i.e. openFT-FTP is not installed.

Other possible values:

YES, i.e. an openFT-FTP license exists, the use of openFT-FTP is activated.

Example

A response file for automatically installing FTAM looks like this:

```
FTAM='YES'
```

```
FTP='NO'
```

2.1.6 Support of Openshift

Starting from version 12.1C10 openFT is prepared to run in constrained environment, such as Openshift. Due to the lack of access to root after generating of Docker image new mode is introduced, called limited multi user mode. Despite its name, it is similar to single user mode. The main difference is that in this new mode, unprivileged user is allowed to create instances. The new "limited multi-user mode", activated via the command "ftsetmode -lm -ga = root: root", may only be used in a containerized environment. An instance created in multi user mode become FT administrator and can perform usual administrative tasks.

Worth to notice that instance is created without systemd support, due to lack of systemd in containerized environment, therefore during building, installing or calling some commands warnings related to it appears.

An exemplary Dockerfile is attached to openFT package in docker/OpenShift location:

```
CD.ftv410.openFT12.1C10_F70.1_20201201.tgz
```

```
|-- docker
|   |-- OpenShift
|   |   |-- Dockerfile
|-- install.ft
|-- openFT-12.1C10-410.x86_64.rpm
```

The Docker file is contained in docker/OpenShift directory and may be similar to the following:

Dockerfile:

```
# Example docker file for OpenShift
# Base system is Redhat8.
FROM registry.access.redhat.com/ubi8/ubi

# Install packages required
RUN yum install openssl procps net-tools tar -y; yum clean all

ENV PACKAGE_DIR /root/openFT_tmp

# switch to temporary directory for openFT package handling
RUN mkdir $PACKAGE_DIR
WORKDIR $PACKAGE_DIR

# Copy the openFT package the root dir. The desired package should be the only one in the
# directory.

COPY CD.ftv*.openFT*.tgz .

# Untar the openFT package to the root dir.
RUN tar -xvf CD.ftv*.openFT*.tgz

# Install openFT.
RUN rpm -i openFT-*.x86_64.rpm

# set workdir back to root's home
WORKDIR /root

# cleanup temporary directory
RUN rm -vfr $PACKAGE_DIR

# Switch to hidden limited multi user mode available under openshift environment.
RUN /opt/openFT/bin/ftbin/ftsetmode -lm -ga=root:root

# Expose basic port used by openFT.
EXPOSE 1100

# Set entry point to the exemplary start script.
ENTRYPOINT "/opt/openFT/bin/ftbin/openft_init"
```

In the example, it runs default `openft_init` (present in the rpm file). The `openft_init` script may be similar to the following:

openft_init:

```
# Export path to the openFT binaries.
export PATH=$PATH:/opt/openFT/bin

# Extract ip address of eth0 interface.
ip_addr=`ifconfig eth0 | grep "inet " | tr -s ' ' | cut -f3 -d ' '`

# Create instance a new user, with eth0 ip address.
ftcrei openft -addr=$ip_addr

# Switch to created instance, to allow usage of openFT for new user.
. ftseti openft

# Start openFT service.
ftstart

# Start /sbin/init as and long time running process.
exec /sbin/init
```

The purpose of the script is to create a proper instance (that will be bound to the eth0 interface ipv4 address, on port 1100) and start the openFT service automatically on each OpenShift's container restart. In the example, /sbin/init has been selected as long running process (process that in case of end or crash will trigger container's restart). Having above structure one can build the Docker image, push it to the OpenShift cluster, and run it. These steps are out of the scope of this document.

Additional notes:

In case of necessity of performing other operations that requires root access (like installing FTAM, FTP, JAVA, or any other), the Dockerfile will have to be expanded, or used as a base. Also exporting other required ports, need to be done during the image build. The user may need to configure file transfer admission.

In recent openFT, usage of systemd was introduced. Under docker/podman build environment the systemd is not available (under container running in OpenShift environment, the case is the same). This may lead to some warning messages during image building process (and can be ignored).

During the ftsetmode command, ftstart is being invoked, which under podman build environment will not work (additional messages will be printed). This is not a problem in case of working container afterwards.

2.1.7 Support of systemd

As of version 12.1C00 of openFT, SysV init is not supported as a method for automatic openFT startup on Linux platform. Instead, systemd is used. During installation of openFT two new systemd units will be installed:

1. openFT.service – this allows automatic startup of all created instances of openFT
2. openFT@.service – this allows selective control on openFT instance

In addition, to customize settings for instances a configuration file `override.conf` is created for each instance.

By default openFT.service is enabled, means it is starting on system startup. During upgrade of existing installation, old SysV init scripts are uninstalled, so any modifications done to them are lost.

As a result, interface of `/etc/init.d/openFT` script is not available anymore and is replaced with following commands:

1. `systemctl start openFT.service` – to start all openFT instances
2. `systemctl stop openFT.service` – to stop all openFT instances
3. `systemctl restart openFT.service` – to restart all openFT instances
4. `systemctl status openFT.service` – shows if openFT has been started with 'systemctl start' since last system startup.
5. `systemctl enable openFT.service` – enables automatic startup of all openFT instances (this is done by default on installation/upgrade)
6. `systemctl disable openFT.service` – disables automatic startup of all openFT instances. Still it is possible to select only some instances to be started automatically
7. `systemctl start openFT@instancename.service` – starts openFT instance, e.g. `systemctl start openFT@std` starts std instance of openFT
8. `systemctl stop openFT@instancename.service` – stops openFT instance
9. `systemctl restart openFT@instancename.service` – restarts openFT instance
10. `systemctl status openFT@instancename.service` – shows status of openFT instance
11. `systemctl enable openFT@instancename.service` – enables openFT instance. Note that enabling openFT service enables also existing instances when this command was executed
12. `systemctl disable openFT@instancename.service` – disables openFT instance. After enabling openFT service followed by disabling of one particular instance, all instances except that disabled one will be started at boot time
13. `systemctl status openFT@*` - shows status of all active instances

As an addition to above list, existing openFT commands have been adopted, so they works as follows:

1. `ftstart` preserves current behavior, but additionally starts openFT@ service of instance in systemd
2. `ftstop` preserves current behavior, but additionally stops openFT@ service of instance in systemd
3. `ftcrei` additionally enables openFT@ systemd service of instance
4. `ftdeli` additionally disables openFT@ systemd service of instance

2.1.7.1 Single-user mode

In general single-user mode is not handled by systemd in any way. Usual openFT commands can be used, as they were before, e.g. `ftstart`, `ftstop`. The only exception is that during instance creation with `ftcrei` command, it is automatically enabled in systemd and on deletion with `ftdeli`, it is disabled.

Remark

OpenFT starts in multi user mode after new installation. Update or correction installation preserves openFT mode. If openFT was running in multi-user mode, it runs in multi-user mode after the update or correction installation. If openFT was running in single user mode, single user mode is retained if openFT is installed again.

2.1.7.2 Enabling ftalarm command

To enable ftalarm automatic startup, one has to edit systemd unit file with 'systemctl edit openFT@instancename' command and uncomment ExecStartPost action starting ftalarm. In case of single-user mode, ftalarm is enabled as it was before, so by editing openFTinst script.

2.1.7.3 Modifying umask value

Similarly to `ftalarm`, to change default umask value, unit file has to be edited with `'systemctl edit openFT@instancename'`, line setting umask value has to be uncommented and desired umask value has to be set.

2.2 Important activities after installation

Following the installation of openFT, you may need to perform additional steps, depending on what you require of your system. These may include the following:

- checking the default settings, see [section “Checking the default settings”](#)
- installing openFT-CR (if encryption of user data is required)
- installing CMX if openFT is to be operated with CMX and CMX was not installed before openFT. You will find the package on the product CD.
- importing configuration data, see [section “Importing configuration data”](#)
- disabling automatic startup of openFT, see [section “Disabling the automatic startup of openFT”](#)
- activating *ftalarm* function, see [section “Enabling the ftalarm command”](#)
- installing or uninstalling openFT-FTAM under HP-UX, AIX and Linux, see [section “Installing or uninstalling openFT-FTAM under HP-UX, AIX and Linux”](#)
- installing or uninstalling openFT-FTP under HP-UX, AIX and Linux, see [section “Installing or uninstalling openFT-FTP under HP-UX, AIX and Linux”](#)
- activating/disabling authentication via PAM (Pluggable Authentication Modules), see [section “Authentication via PAM”](#)
- creating the partner list from TNS, see [section “Creating the partner list from TNS”](#)
- configuring the remote administration server
If you want to use your system as a remote administration server, you must configure the server. See [section “Configuring the remote administration server”](#).
- configuring the ADM trap server
If you want to use your system as an ADM trap server, you must configure the server. See [section “Configuring the remote administration server”](#).
- creating TNS entries
If you use the TNS you may need to create the TNS entries, see [section “Entering transport system applications in the TNS”](#).

If no or no current TNS entries are present for openFT V12 (because CMX was installed after openFT), then you can subsequently create or update these using a script, see [section “Entering transport system applications in the TNS”](#).

i Please note that cluster configurations are only supported for TCP/IP. You are therefore recommended to work without CMX and TNS.

i As of openFT version V12.1B, the release is only for Linux x86_64. Since openFT version V12.1C20, Solaris (Sparc) is also included in the release again. Further platforms are available on request from your Fujitsu sales representative.

2.2.1 Checking the default settings

In the case of a new installation, openFT sets default values for the operating parameters and FTAC settings. These are chosen in such a way that they generally suffice for normal openFT operation. However, you should check whether these settings are suitable for your particular application and requirements. The special functions such as remote administration server, trace, traps, automatic deletion of log records etc. as well as the use of TNS and CMX are deactivated.

The default admission set is defined in such a way that unrestricted file transfer is possible. As FTAC administrator you should therefore modify the standard admission set to match the security needs of the computer (see also [section "Administering admission sets"](#)).

Operating parameter settings

Following a new installation (including the installation of openFT-FTAM, openFT-FTP and openFT-CR), you can use the *ftshwo* command to display the settings:

```
ftshwo
STARTED PROC-LIM CONN-LIM ADM-CLIM RQ-LIM MAX-RQ-LIFE TU-SIZE CCS-NAME
  YES      NONE      16      8      2000      30      65535      ISO88591
PTN-CHK DYN-PART SEC-LEV FTAC-LOG FT-LOG FT-DIR-LOG ADM-LOG USE TNS USE CMX
  STD      ON      B-P-ATTR  ALL    ALL      NONE    ALL      NO    NO
OPENFT-APPL FTAM-APPL FTP-PORT ADM-PORT ADM-CS
*STD      *STD      21      11000      NO
ACTIVE    ACTIVE    ACTIVE    ACTIVE
RSA-PROP RSA-MIN AES-MIN ENC-MAND
2048      0      NONE    NO
HOST-NAME IDENTIFICATION / LOCAL SYSTEM NAME
*NONE    mc011.mynet.local / $FJAM,MC011
FT-ADMIN FTAC-ADMIN
ROOT      *STD
FN-CSS-NAME DEL-LOG ON AT RETPD RECOVERY ADM-TRAP-SERVER
  ISO88591 OFF DAILY 00:00 14 IN+OUT *NONE
TRAP: SS-STATE FT-STATE PART-STATE PART-UNREA RQ-STATE TRANS-SUCC TRANS-FAIL
CONS OFF OFF OFF OFF OFF OFF OFF OFF
ADM OFF OFF OFF OFF OFF OFF OFF OFF
FUNCT: SWITCH PARTNER-SELECTION REQUEST-SELECTION OPTIONS OPTIONS-LL
MONITOR OFF ALL ALL
TRACE OFF ALL ALL NONE OFF
```

For a detailed description of the individual values, see the manual "openFT (Unix and Windows systems) - Command Interface".

You should first of all check the following settings:

- Instance ID

This is preset to the name of the computer in the local network. If this is not sufficient to ensure unambiguous identification throughout the network, please change the value (*ftmodo -id*), see also [section "Unique Instance Identifications"](#).

- Local port numbers for the protocols (OPENFT-APPL, FTAM-APPL, FTP-PORT).

If you use other port numbers for addressing, e.g. for openFT, FTAM or FTP, then you should adapt these (*ftmodo* command, options *-openft*, *-ftam* and *-ftp*).

- Dynamic partners (DYN-PART)

Dynamic partners are permitted. If you want to lock this for security reasons then use `ftmodo -dp=f` to set the value to OFF.

You can also change the operating parameter settings in the openFT Explorer. To do this, open the *Administration* menu and choose the *Operating Parameters - Options* command.

FTAC settings

Following a new installation, all the values for the default admission set are set to 100. This means that the system is open for all users with a valid password, for all partners and for all actions. You should adapt the default admission set to meet the security needs of the system. You can do this using the `ftmoda` command, see the manual "openFT (Unix and Windows systems) - Command Interface". Alternatively, you can use the openFT Explorer and adapt the *STD admission set in the *Admission Sets* object window.

2.2.2 Importing configuration data

You can import configuration data that you have created in another system, for example by means of an export:

- You import operating parameter settings by executing the export file in the shell.
- You import the FTAC environment by means of the *ftimpe* command or in the openFT Explorer by opening the *Administration* menu and choosing the *FTAC Environment - Import FTAC Environment* command.
- You import a partner list by executing the export file in the shell.
- You import the configuration for a remote administration server by means of the *ftimpc* command or in the openFT Explorer by opening the *Administration* menu and choosing the *Remote Administration - Import Configuration* command.

2.2.3 Disabling the automatic startup of openFT

i This section only applies to openFT versions \leq V12.1C, since from version V12.1C systemd is used.

i This section does not apply to Solaris, because openFT V12.1 does not support automatic start-up of openFT on Solaris without SMF. A different mechanism is used for Solaris with SMF, see [section “Solaris SMF”](#).

During installation, the startup file is installed, e.g. `/sbin/rc2.d/S910openFT` on HP-UX, `/etc/rc3.d/S50openFT` on RedHat Linux and `/etc/rc.ft` on AIX. This script calls the file `/var/openFT/std/etcinit/openFTinst` when the system starts, which then automatically starts openFT.

If openFT instances were created using the `ftcrei` command, then this script also calls the startup and shutdown file for this instance (see also [section “openFT instances and cluster operation”](#)).

These files then start openFT on each openFT instance.

If you do not want openFT to be started automatically, you will need to comment out the corresponding command line in the file `/var/openFT/std/etcinit/openFTinst` or in the startup and shutdown file for the instances.

Automatic termination of openFT

During installation, the shutdown file is installed (e.g. `/sbin/rc1.d/K200openFT` on HP-UX or `/etc/rc0.d/K50openFT` on RedHat Linux or `/etc/rc.ft` on AIX). This script calls the file `/var/openFT/std/etcinit/openFTinst` when the system shuts down, which then automatically terminates openFT.

If openFT instances were created using the `ftcrei` command, then this script also calls the startup and shutdown files for these instances (see also [section “openFT instances and cluster operation”](#)).

These files then terminate the corresponding openFT instance.

2.2.4 Enabling the *ftalarm* command

i This section does not apply to Solaris where a different mechanism is used, see [section “Solaris SMF”](#).

If you want to be informed about the frequency of failed FT requests, it is advisable to use the *ftalarm* command for this purpose see the manual "openFT (Unix and Windows systems) - Command Interface".

If desired, you can also have the *ftalarm* command automatically started at system startup by inserting a corresponding line with the *ftalarm* command in the startup and shutdown file `/var/openFT/std/etcinit/openFTinst` and `/or` in the startup and shutdown files of other instances.

2.2.5 Solaris SMF

i As of version openFT V12.1B, the platform Solaris is only supported on request.

SMF (Service Management Facility) can be used to describe in detail the dependencies of a service on other services, files or milestones (correspond to the earlier run levels), as well as instances of the service, in a manifest.

This results, for example, in significantly shorter start times because many services can be started in parallel and the start sequence can be optimized thanks to the description of the dependencies.

The various services in the system are administered via a uniform interface. This ensures that operation is more robust, i.e., for example, if a service is terminated unexpectedly (e.g. due to an unintentional *kill -9*) then it is restarted automatically.

Operation with SMF differs from operation without SMF as follows:

- The start scripts `/etc/init.d/openFT` and `/var/openFT/instance/etcinit/openFTinst` are **not** installed with Solaris SMF. *ftalarm* is activated via SMF, see [section “Solaris SMF”](#).
- There is no automatic check of the profile files and no automatic clean-up of the log files.
- The dependency on CMX is not defined since CMX is not involved in the SMF installation procedure. If openFT is used with CMX then diagnostic records may be generated for openFT during booting. These are created during the period before CMX becomes available. The reason is that all the SMF milestones are first activated and that the RC scripts are then started. This means that CMX is not started until after openFT. In openFT V12.0 or higher, it is no longer essential for CMX to be present when RFC1006 is used.

The following commands have been adapted for use with the SMF procedure to ensure that they function in the usual way:

- *ftstart* takes over environment variables and starts openFT via SMF. The SMF command (without transfer of environment variables) is as follows:

```
svcadm enable openFT:instance
```

The familiar openFT messages are not displayed for *svcadm*.

- *ftstop* terminates openFT via SMF. The SMF command is as follows:

```
svcadm disable openFT:instance
```

The familiar openFT messages are not displayed for *svcadm*.

- In addition to the instance, *ftcrei* also generates a manifest and enters this in SMF.
- *ftdeli* deletes the instance and removes the corresponding manifest from SMF.

i For the setting of protection bits using SMF, please refer to [section “Setting the protection bit for newly created files \(Unix systems\)”](#).

Characteristics in single-user mode

In single-user mode the openFT instances that are enabled in the SMF are not monitored by the SMF, i.e. the openFT user can start or stop the openFT instance at will with the commands *ftstart* and *ftstop*. Only the system administrator (*root*) can use the SMF commands to define whether the openFT instance is started (enabled) or not started (disabled), see above.

When booting the system the settings defined by the system administrator apply.

When switching to multi-user mode *ftstart* and *ftstop* are also used to execute the corresponding *svcadm* command.

Activating *ftalarm* automatically

As described below, *ftalarm* can be started and stopped manually via the command line. On Solaris systems, *ftalarm* can also be administered via SMF. The *ftalarm* manifest that is required for this is automatically generated and installed for each instance.

i Mixed operation (manual operation and control via SMF) is not recommended because SMF is not informed of changes. As far as SMF is concerned, *ftalarm* is a so-called transient service, i.e. there is no process to be monitored.

Example

ftalarm can be started for the instance *inst001* using the following commands:

```
# svcadm enable ftalarm:inst001
# svcadm disable ftalarm:inst001
```

Creating the instance *inst001*:

```
# svcadm enable ftalarm:inst001
# svcadm disable ftalarm:inst001
```

Creating the instance *inst001*:

```
# ftcrei 001 -addr=inst001
# svcs *:inst001
STATE          STIME          FMRI
disabled       16:31:50      svc:/application/openFT:inst001
disabled       16:31:51      svc:/application/ftalarm:inst001
# svcadm enable ftalarm:inst001
# svcs *:inst001
STATE          STIME          FMRI
disabled       16:31:50      svc:/application/openFT:inst001
online         16:32:14      svc:/application/ftalarm:inst001
#. ftseti inst001
# ftstart
ftstart: openFT 12.1A00 starting. Protocols: openFT,FTAM,ADM
```

```
# svcs *:inst001
STATE          STIME          FMRI
online         16:32:37      svc:/application/openFT:inst001
online         16:32:38      svc:/application/ftalarm:inst001
```

The *ftalarm* cronjob for the instance *inst001* is not started unless the instance *inst001* has also been started. Similarly, the instance *ftalarm* is terminated when the instance *inst001* is terminated with *ftstop*.

The number of errored FTAC sets can be set using the *ftalarm* instance's SMF environment variable *ERRORS*, e.g. as follows for the instance *inst001*:

-
1. Terminate *ftalarm* for the instance *inst001* using the command:

```
# svcadm disable ftalarm:inst001
```

2. Change the number of errors for monitoring (e.g. to 42) using the command:

```
# svccfg -s ftalarm:inst001 setenv -i ERRORS 42
```

3. Take over the settings using the following command:

```
# svcadm refresh ftalarm:inst001
```

4. Start *ftalarm* for the instance *inst001* using the command:

```
# svcadm enable ftalarm:inst001
```

5. You can display the settings with:

```
# svcprop -t -p method_context/environment ftalarm:inst001  
method_context/environment astring OPENFTINSTANCE=inst001 ERRORS=42
```

2.2.6 Installing or uninstalling openFT-FTAM under HP-UX, AIX and Linux

i As of openFT version V12.1B, the release is only for Linux x86_64. Since openFT version V12.1C20, Solaris (Sparc) is also included in the release again. Further platforms are available on request from your Fujitsu sales representative.

As of openFT version V12.1C30, `install.ftam` and `install.ftp` do not exist anymore, because they are replaced with the license system. Running `install.ftp` and `install.ftam` on openFT V12.1C30 or later will have no effect

openFT-FTAM is not installed together with openFT when the installation is a new installation on an HP, AIX or Linux system. The same applies to patch installations when openFT-FTAM was not installed beforehand.

In these cases you need to install openFT-FTAM using the `install.ftam` command after installing openFT. You will find this command in the directory `/opt/openFT/bin/ftbin`, see also the manual "openFT (Unix and Windows systems) - Command Interface".

Installation is only permitted if a valid openFT-FTAM license is available.

You can also uninstall openFT-FTAM if it is not needed anymore using `install.ftam`. openFT-FTAM must be uninstalled if you do not have the corresponding license.

2.2.7 Installing or uninstalling openFT-FTP under HP-UX, AIX and Linux

i As of openFT version V12.1B, the release is only for Linux x86_64. Since openFT version V12.1C20, Solaris (Sparc) is also included in the release again. Further platforms are available on request from your Fujitsu sales representative.

As of openFT version V12.1C30, `install.ftam` and `install.ftp` do not exist anymore, because they are replaced with the license system. Running `install.ftp` and `install.ftam` on openFT V12.1C30 or later will have no effect

openFT-FTP is not installed together with openFT when the installation is a new installation on an HP, AIX or Linux system. The same applies to patch installations when openFT-FTP was not installed beforehand.

In these cases you need to install openFT-FTP using the `install.ftp` command after installing openFT. You will find this command in the directory `/opt/openFT/bin/ftbin`, see also the manual "openFT (Unix and Windows systems) - Command Interface".

Installation is only permitted if a valid openFT-FTP license is available.

You can also uninstall openFT-FTP if it is not needed anymore using `install.ftp`. openFT-FTP must be uninstalled if you do not have the corresponding license.

2.2.8 Authentication via PAM

i As of openFT version V12.1B, the release is only for Linux x86_64. Since openFT version V12.1C20, Solaris (Sparc) is also included in the release again. Further platforms are available on request from your Fujitsu sales representative.

PAM (Pluggable Authentication Modules) consists of a collection of program libraries which allow system administrators to choose the way applications authenticate users. openFT supports the PAM interface for user authentication in the operating systems Linux, Solaris, HP-UX and AIX.

Following installation, the PAM function is enabled on Linux, Solaris and HP-UX systems but is disabled on AIX systems. Under AIX, you must therefore enable the PAM function explicitly, see [section “Authentication via PAM”](#).

In many cases, it is necessary to check the configuration files and adapt the entries, see [section “Authentication via PAM”](#).

Enabling/disabling the PAM function

At runtime, you can enable or disable the PAM function on all platforms using the environment variable OPENFTPAM. To do this, you must stop the asynchronous openFT server (e.g. with the *ftstop* command), set the variable and then restart the asynchronous openFT server (e.g. with the *ftstart* command):

```
OPENFTPAM=ON
```

```
export OPENFTPAM
```

PAM function is enabled.

```
OPENFTPAM=OFF
```

```
export OPENFTPAM
```

PAM function is disabled.

Checking and modifying the PAM configuration files

The PAM mechanism is controlled by means of application and platform-specific configuration files.

- Linux

On Linux, the PAM mechanism is controlled by means of files in the directory */etc/pam.d* or by means of an entry in the file */etc/pam.conf* if */etc/pam.d* does not exist.

When logging on to PAM, openFT uses the service name *openft*. In the case of an openFT update installation /new installation, a configuration file with the name *openft* is therefore created in the directory */etc/pam.d* if no such file already exists. The authentication mechanism that is to be used is defined in this file. If the system administrator has defined a specific authentication mechanism via the file */etc/pam.d/common-auth* then this is used by openFT. If not, the PAM module *pam_unix.so* for user authentication under Linux is used.

If the directory */etc/pam.d* does not exist then the system administrator must make a suitable entry in the file */etc/pam.conf* for the service name *openft*.

- Solaris, HP-UX and AIX

The PAM mechanism functions on these platforms for openFT if the file `/etc/pam.conf` contains an entry for OTHER with service module type `auth` which permits the applications installed on the relevant operating system to use the PAM functionality. If this is not the case then you must make the following entry in the file `/etc/pam.conf`:

- Solaris

Depending on your Solaris version, you may need to make the following entries:

```
openft auth required pam_unix.so.1
openft auth requisite pam_authtok_get.so.1
openft auth required pam_unix_auth.so.1
```

- HP-UX

```
openft auth required libpam_unix.1
```

and if necessary also

```
openft auth required libpam_unix.so.1
```

- AIX

On AIX systems, it is possible that the entry for OTHER is configured as follows by default and therefore prohibits the service:

```
OTHER auth required pam_prohibit
```

In this case, it is necessary to make the entry for openFT separately:

```
openft auth required pam_aix
```

2.2.9 Creating the partner list from TNS

i On Unix systems, PCMX is supported and provided with openFT V12.1B for the last time. Please switch all TNS entries concerning openFT partners to partner list entries.

Due to the use of the partner list, openFT makes it possible to work without TNS. Compared to TNS, the partner list has the advantage that you can use it to store not only all the necessary address information but also other properties such as, for example, a partner's security level.

If you want to switch to operation without TNS then you can use the tool *tns2ptn*. *tns2ptn* is used to create new partner list entries on the basis of TNS entries with the RFC1006 address format.

You must perform the following steps to insert TNS entries in the partner list:

1. Export the TNS entries to a file:

To do this, enter the command `tnsxprop > openft.tns` (where *openft.tns* is the file name that you can choose yourself).

2. If necessary, clean up the export file (here *openft.tns*) by deleting the entries that do not relate to openFT, are no longer required or do not have the RFC1006 address format.
3. Call the tool *tns2ptn*:

```
/opt/openFT/bin/ftbin/tns2ptn openft.tns > ft_list
```

ft_list is the name of the output file and can be selected freely. *ft_list* contains an *ftaddptn* command with the associated address information for each partner.

If an entry cannot be converted then it is output at *stderr*.

4. Run the output file (here *ft_list*) as FT administrator at command level (e.g. *sh ft_list*).

Please note that the address information is taken over from TNS. Additional partner properties (security level, priority, tracing etc.) can subsequently be defined using the *ftmodptn* command or via the openFT Explorer.

3 Installation on Windows systems

This chapter describes the installation and configuration of openFT on Windows systems.

i openFT is shipped with a communications manager (PCMX).

3.1 Installation of openFT

The installation of openFT is performed under a user id with Windows administrator rights.

openFT V12.1 is installed using Microsoft's Windows Installer. Start guided installation in Windows in the normal manner by double-clicking the *setup.exe* program located on the data medium containing the openFT software. You can also install openFT in "unattended" mode. See the [section "Unattended installation"](#).

There are three different types of installation depending on whether an FT version is already installed or which FT version is already installed on your computer:

- New installation
Your computer does not yet have an openFT or has an openFT < V11.0 on it.
- Update installation
Your computer has openFT version V11.0 or V12.0 installed.
- Installation of a correction version (patch)
Your computer has openFT version 12.1 installed.

You can extend an existing installation of openFT using the Change function in Windows and install or uninstall the openFT SNMP agents at a later date. In Windows 10, for instance, the function is located under *Control Panel - Programs and Functions - openFT - Repair*. This function also allows you to repair an existing installation if necessary.

What you need to observe before installing openFT ...

- If *German* or *English* is set as the language in the operating system then this language is used during installation and is set by default as the language for openFT. In the case of all other system languages, you are asked whether German or English is to be used for installation and as the default language (see also [section "Switching the language interface"](#)).
- A Windows user who wants to be able to work with openFT requires the right "Log on locally".
- For openFT versions <= V12.1B:
If you want to encrypt file contents, you must also install openFT-CR V12.1 (Windows). This software is offered without a license at a fixed price. If an openFT-CR version < V11.0 is already installed, then you must first uninstall this version before installing openFT. You may only install openFT-CR V12.1 after openFT V12.1 has been installed.



- Following a new installation of openFT V12.1, only openFT-CR V12.1 can be used. Older versions of openFT-CR from V11.0 or V12.0 can no longer be installed following a new installation of openFT V12.1.
- The following applies following an update installation of openFT V12.1:
If openFT-CR from V11.0 or V12.0 was previously installed then the openFT-CR functionality is already activated and remains present following the update installation. If you subsequently uninstall the old openFT-CR then you can no longer reinstall the old openFT-CR.

- If you want to use the openFT-Script interface or the Java API then the J2SE™ Runtime Environment 7.0 (JRE 7.0 with update 25) or higher must be installed on your system.
- Installation of the SNMP support requires an installed Microsoft SNMP Server, see [section "Installation of the SNMP subagent"](#).

- Installation directory

The path under which openFT is installed depends on a number of factors and is generally referred to below as the *openFT-installation-directory*.

The following points apply:

- The path depends on your operating system. By default, openFT is installed in the directory *%ProgramFiles%\openFT*.
- During installation, you can freely specify the installation directory. You must not, however, specify a network drive as the installation path. The maximum path length is 110 characters.

It is recommended that you use the suggested path.

- Instance directory

The instance directory is set up during installation and contains subdirectories for application-specific data for the corresponding openFT instance, such as the log file, key pair sets and trace files. On Windows systems, the default pathname is

%ProgramData%\Fujitsu Technology Solutions\openFT\var\instance.

instance is the name of the corresponding instance. The standard instance named *std* always exists.

i When you create a new instance using *ftcrei*, you can select any path name for the instance directory.

The following sections describe which steps must be performed for the three installation variants by you as the system administrator as well as those which are handled automatically by the installation procedure.

3.1.1 New installation

If you have not yet installed any version of openFT on your computer or if a version < V11.0 is installed, the installation is a new installation.

Tasks required of the system administrator

1. If openFT version 10.0 and possibly add-on products are already installed, then you should proceed as follows:
 - Save admission profiles and admission sets that are still needed in an external file using *ftexpe*.
 - Save the partner list entries using *ftshwptn* in an external file and - if necessary - the openFT operating parameter settings using *ftshwo*.
 - Uninstall openFT-CR, openFT and the add-on products.
2. Install the openFT V12.1 product software.

During installation, you are asked to enter the license keys:

- If you specify a valid basic license key then the basic functionality of openFT will be enabled (openFT protocol, FTADM protocol and FTAC functionality). The optional license keys for the FTAM and FTP protocols are then queried. If you enter a valid license key for an optional protocol then this protocol is also enabled. You can skip this dialog box if you do not possess a valid key for FTP or FTAM.

i If you enter a valid license key for a special version (e.g. CL or FTAM-CL version) then you will not be asked to enter a license key for FTAM and/or FTP because the supported function scope is already present in the special version.

- If you do not enter a license key then openFT runs with full functionality as a demo version for 30 days. This demo version may only be used for evaluation purposes!

Following installation, you can enter openFT license keys at any time using the openFT Explorer (*Administration - License Administration*) or the *ftaddlic* command.

The identification (*ftmodo -id=*) are set automatically during initial installation. You should check that these values are correct.

3. Import the saved configuration data as follows, if necessary:
 - Import the saved admission sets and admission profiles using *ftimpe*. If the admission sets and admission profiles were exported from an openFT version < V8.1 then all security levels in the admission sets that were previously set at 1 are automatically converted to 90. The standard admission set is re-set.
 - Import the saved partner list entries and operating parameter settings (if applicable) by executing the save file in the command prompt.

After these steps, openFT will be fully operational and will be activated at each system startup.

Steps performed automatically

During installation, the following steps are carried out automatically:

- If CMX is installed then default TNS entries are generated for openFT if no TNS entries yet exist, otherwise they are adapted (see [section "TNS entries created automatically"](#)).

If CMX is subsequently installed then you can also use a tool to create default TNS entries at some later date, see [section "Entering transport system applications in the TNS"](#).

-
- The instance directory for the standard instance is set up, see [section “Installation of openFT”](#). The instance-specific files are located in the directory

%ProgramData%\Fujitsu Technology Solutions\openFT\var\std.

In this case, the operating parameters (e.g. maximum number of requests that can be processed simultaneously, maximum block length, scope of FT and FTAC logging, setting of the CCS, port numbers for the asynchronous inbound servers) are set to default values, see also [section “Checking the default settings”](#).

CMX operation, FTP server and the use of the TNS are deactivated.

- The name of the processor is entered as the processor name.
- The DNS name of the computer (if one exists) is pre-set as the instance ID for the standard instance. When there is no DNS name, the name of the computer is used for the instance ID.
- A standard admission set permitting all file transfer functions is created.
- A key pair set is created (see [section “Creating and administering local RSA key pairs”](#)).
- The openFT service and the asynchronous openFT server are started.

3.1.2 Update installation of openFT

If openFT V11.0 or V12.0 is already installed, an update installation is performed.

Points to observe preparatory to an update installation

During an update installation, the following actions are carried out for all active instances including the standard instance:

- The log file is deleted. Therefore you should evaluate the log records before performing the update installation.
- Any running openFT-Script requests are aborted during installation. All old, aborted openFT-Script requests are not regarded as being restartable in the new openFT version. You should therefore complete all running openFT-Script requests before carrying out an update installation.
- Existing trace files, if any, and diagnostics files are deleted.

If you wish to continue using openFT instances that have been deactivated using *ftdeli*, you should activate them before the update installation using *ftcrei*. The corresponding instance file trees are then automatically updated during installation. If you do not do this, you must update these instances after installation using the *ftupdi* command (see the manual "openFT (Unix and Windows systems) - Command Interface").

Tasks required of the system administrator

1. Install openFT from the data medium.
2. During installation, you are prompted to enter the license key.

Following an update installation, the asynchronous openFT server for the standard instance is started automatically, i.e. the operating parameter option *Start Asynchronous Server Automatically* is activated in the openFT Explorer. To access this option, choose the *Operating Parameters* command in the *Administration* menu and go to the *General* tab. This option is deactivated for the other instances. You must therefore activate this option in these instances if you always want the associated asynchronous openFT server to be started automatically.

Steps performed automatically

The following steps are performed automatically for an update installation:

- openFT-Script requests are cancelled.
- In the case of an openFT V10.0 update installation, the default TNS entries for openFT are handled as follows:
 - Default TNS entries from older openFT versions < V10.0 that are no longer required are deleted
 - Missing required default TNS entries are created.
 - Existing required default TNS entries remain unchanged.

-
- The language setting from the previous version is used. The instance directories of currently existing instances including the standard instance are updated, i.e.:
 - The log file is deleted.
 - During this, the following configuration data are used:
 - Operating parameters (the operation CMX with remains activated)
 - Instance identification
 - partner list entries
 - Admission sets and profiles:
 - Key pair sets:
 - Configuration data for central administration (in the case of an update from V11.0).

If it is necessary to reboot the computer on an update installation then the instance update is not performed until after the reboot. If errors occur, a *ftupdi-instance.log* file with corresponding error messages is created in the `%ProgramFiles%\openFT` directory. After each update installation with reboot you should check whether there are *ftupdi-instance.log* files in the `%ProgramFiles%\openFT` directory. If so, the instances must be updated manually using the *ftupdi directory* command. The associated *ftupdi-instance.log* file can be deleted after a manual update.

i As of openFT version 12.1C20, there are two additional key pairs with lengths 3072 and 4096. They are generated after the update installation is complete as soon as the instance is running for the first time.

3.1.3 Installation of a patch

Installation of a patch means that openFT V12.1 is already installed on your computer. Please note the following:

- Any running openFT-Script requests are aborted during installation. You should therefore complete all running openFT-Script requests before installing a correction version.
- Any trace files, diagnostic records or files with console commands that may be present are deleted.

Tasks required of the system administrator

1. Install openFT V12.1 from the data medium.

All license keys are taken over and no confirmation query is issued. The installation status of the SNMP subagent (installed/not installed) remains unchanged.

Steps performed automatically

The following steps are performed automatically on installing a patch:

- Running openFT processes are terminated, running openFT-Script requests are cancelled.
- The admission profiles and admission sets, the log files, operating parameters and requests, the partner list, the key pair sets, and the configuration data for the central administration are taken over without changes for all openFT instances.
- The language setting from the previous version is used.
- The configuration data for the central administration is used.

3.1.4 Unattended installation

CMX and openFT can also be installed unattended using the `msiexec` command (Windows Installer). Please note the following:

- Installation must be started from a console with administrator rights, i.e. the installation must either be performed by the administrator or the console must have been started with *Run As Administrator* with user account control activated. Otherwise you will not be authorized to perform the installation!
- If you want to work in "operation with CMX" mode then you should install CMX before installing openFT. This automatically generates the default TNS entries for openFT.

You use `msiexec` to install the MSI installation packages `openFT.msi` and/or `PCMX-32.msi`. Both of these are located in the directory `openFT\Unattended_installation` on the product CD.

When calling `msiexec`, specify the relevant MSI file as an argument.

Possible other Windows Installer parameters:

ADDLOCAL:

ADDLOCAL is used to specify which optional features are installed. Possible value is for ADDLOCAL is SNMP.

ADDLOCAL SNMP means that the openFT SNMP subagent is installed in addition to openFT. For this to be possible, the Microsoft SNMP service must be installed first, see [section "Installation of the SNMP subagent"](#).

If ADDLOCAL is not specified then only the openFT protocol is installed whereas SNMP support is not installed.

openFT can only use transfer protocols for which a valid license key is available (see the openFT properties LICENSEKEY, FTAMLICENSEKEY and FTPLI-CENSEKEY).

TRANSFORMS:

The German variant of openFT can be set by specifying the TRANSFORMS:de parameter. For details on setting the language see [section "Switching the language interface"](#).

openFT properties:

LICENSEKEY:

A valid openFT license key must be specified for the LICENSEKEY parameter. If you do not enter a license key then openFT runs with full functionality as a demo version for 30 days. This demo version may only be used for evaluation purposes!

Following installation, you can enter the openFT license key at any time using the openFT Explorer (*Administration - License Administration*) or the `ftaddlic` command.

The LICENSEKEY consists of five subkeys of five characters each and separated from one another by a minus sign.

Example:

12345-12345-12345-12345-12345

FTAMLICENSEKEY:

The specification of FTAMLICENSEKEY is optional.

A valid license key for the FTAM protocol must be specified for FTAMLICENSEKEY. The specification of FTAMLICENSEKEY is only of use if you have entered a license key for a server version of openFT in LICENSEKEY.

If you have not specified a license for the FTAM protocol, you should not specify FTAMLICENSEKEY. In this case, the FTAM protocol cannot be used. Following installation, you can enter the license key for the FTAM protocol at any time using the openFT Explorer (*Administration - License Administration*) or the *ftaddlic* command.

The FTAMLICENSEKEY consists of five subkeys of five characters each and separated from one another by a minus sign.

Example:

54321-54321-54321-54321-54321

FTPLICENSEKEY:

The specification of FTPLICENSEKEY is optional.

A valid license key for the FTP protocol must be specified for FTPLICENSEKEY. The specification of FTPLICENSEKEY is only of use if you have entered a license key for a server version of openFT in LICENSEKEY.

If you have not specified a license for the FTP protocol, you should not specify FTPLICENSEKEY. In this case, the FTP protocol cannot be used. Following installation, you can enter the license key for the FTP protocol at any time using the openFT Explorer (*Administration - License Administration*) or the *ftaddlic* command.

The FTPLICENSEKEY consists of five subkeys of five characters each and separated from one another by a minus sign.

Example:

32154-32154-32154-32154-32154

INSTALLDIR:

The INSTALLDIR parameter enables you to specify the installation directory for openFT, see also [section "Installation of openFT"](#).

The INSTALLDIR parameter can also be specified with the PCMX-32.msi package for unattended CMX installation.

By default, openFT is installed in the directory *%ProgramFiles%\openFT*.

You must not specify network drive paths or UNC paths as the installation path (INSTALLDIR).

openFT must be installed on a local hard disk.

The maximum length of the selected installation directory is 110 characters. If you specify a longer installation directory then installation is rejected.

It is recommended that you omit the INSTALLDIR parameter. In this case, the default installation path is used.

Examples

1. You start unattended installation of CMX without user interaction with

```
msiexec /i PCMX-32.msi /qn
```

2. You start unattended installation of the German language version of openFT (without user interaction) in the default directory `%Program Files%\openFT` with

```
msiexec /i openFT.msi TRANSFORMS=:de /qn
```

3. Enter the following command to start unattended installation without user interaction of the German language version of openFT including SNMP in the default directory:

```
msiexec /i openFT.msi ADDLOCAL=SNMP TRANSFORMS=:de /qn
```

4. For the unattended installation, without user interaction, of openFT with SNMP in German in the default directory including specification of the openFT and optional FTAM license keys, enter the following command:

```
msiexec /i openFT.msi ADDLOCAL=SNMP  
LICENSEKEY=12345-12345-12345-12345-12345  
FTAMLICENSEKEY=54321-54321-54321-54321-54321  
TRANSFORMS=:de /qn
```

i If you select unattended installation (e.g. option `/qn` or `/qb`), the system is restarted automatically by the Windows Installer if necessary. You can suppress restart by setting the `/norestart` option (Windows Installer 3.0 or higher) or the parameter `REBOOT=ReallySupress`.

In the case of unattended installation, the exit code of `msiexec.exe` indicates whether installation was successful or not. To use this facility, start `msiexec` as follows:

```
start /wait msiexec /l*vx install.log /i openFT.msi  
ADDLOCAL=SNMP  
LICENSEKEY=12345-12345-12345-12345-12345  
FTAMLICENSE=54321-54321-54321-54321-54321  
TRANSFORMS=:de /qn /norestart
```

Directly after completion of the `msiexec` command, use `echo %ERRORLEVEL%` to query the exit code:

- 0 Installation successful, it is not necessary to restart the computer
- 3010 Installation successful, it is essential to restart the computer

All other exit codes (e.g. 1603) indicate an error during installation, for example the specification of an invalid license key or an invalid installation directory.

If an error occurs then the reason can be determined via the optional installation log `install.log`.

In the example above, this logging was activated using the `//*vx install.log` option in the `msiexec` command at the start of installation. At the end of the logged installation sequence and before the concluding list of properties, it is usually possible to find the error message and a description of the error that caused the installation operation to be aborted.

A detailed description of the `msiexec` command together with a complete list of the possible exit codes can be found on the Microsoft web site.

3.1.5 Installation of the SNMP subagent

In order to install the openFT sub agent on Windows systems, the SNMP Service from Microsoft must be installed first.

You can then install the openFT subagent.

i It is only possible to manage one openFT instance using SNMP. This is the instance that was set with the system environment variable OPENFTINSTANCE before the SNMP service was started. This variable is not set by default. In this event, the instance *std* is administered.

Installing/activation the SNMP service

This section describes installation using Windows 10 as an example. Installation on other Windows systems may differ slightly. For further information, refer to the Windows documentation.

Under Windows 10 proceed as follows:

- From the Control panel select *Programs and Functions*.
- In the *Add or Remove Programs* dialog press the *Enable/Disable Windows Functions* button. In the next dialog select *Simple Network Management Protocol (SNMP)* from the list of components.
- After installation the SNMP service it must be configured by selecting SNMP Service under *Services* in the Control Panel. It is possible to specify your own name under *Contact* and your location under *Location* in the index card *Agent*. Nothing need be entered in the index card *Traps*, while a Community string should be entered in the *Security* index card. The Community string acts as a password. In the subsequent administration of the sub agent, this string functions as a password. In addition, access rights must still be assigned. If it is likely that during subsequent openFT operation settings will not only be read but also modified (e.g. in order to start or terminate openFT), it is advisable to select *READ WRITE* at this point. Otherwise the specification *READ ONLY* would be quite sufficient.

Installing the openFT sub agent

Proceed as follows:

- From the Control Panel select *Add or Remove Programs*.
- Select *openFT V12.1A00* from the list of currently installed programs and press the *Change* button.
- In the dialog *Application Maintenance* select the *Modify* option and press the *Next >* button.
- In the *Select Features* dialog select the *SNMP agent* feature for installation and press the *Next >* button. After confirming the selection the openFT SNMP sub agent will be installed.
The MIB Management Information Base file *openFTMIB.txt* will be stored in the directory *openFT-installation-directory\snmp*.
- After installation of the SNMP sub agent the SNMP service must be restarted.

3.1.6 Changing and repairing openFT

An installed openFT version can be changed and repaired using the normal Windows tools.

Before performing any change or repair operations, you should close the openFT Explorer. Otherwise, it may be necessary to restart the computer because certain files could not be replaced.

Other services, such as the asynchronous openFT server, for example, are automatically terminated on change or repair operations.

If the installation is changed or repaired, all the settings are retained.

3.1.7 Deinstallation of openFT, openFT-CR and CMX

When uninstalling, please note the following:

- You can uninstall the optional products openFT-CR and CMX individually and independently of openFT.
- If you want to uninstall CMX, you must first deactivate operation with CMX, for example using the command *ftmodo -cmx=n*.
- If you want to uninstall openFT on a computer on which openFT-CR is installed, you must first uninstall openFT-CR.
- The software can be uninstalled using the *Control Panel*. Depending on the operating system, choose *Programs and Functions* or *Programs - Programs and Functions*.

3.2 Important activities after installation

Following the installation of openFT, you may need to perform additional steps, depending on what you require of your system. These may include the following:

- checking the default settings, see [section “Checking the default settings”](#)
- entering the user password (*ftsetpwd* command or in the openFT Explorer, *Administration* menu, *User Password...* command)
- installing openFT-CR (if encryption of user data is required)
- installing CMX if openFT is to be operated with CMX and CMX was not installed before openFT. You will find the corresponding file *PCMX-32.msi* on the product CD in the directory *openFT\Unattended_installation*.
- importing configuration data, see [section “Importing configuration data”](#)
- activating/disabling automatic startup of openFT, see [section “Activating and disabling the automatic startup of openFT”](#)
- creating the partner list from TNS, see [section “Creating the partner list from TNS”](#)
- configuring the remote administration server
If you want to use your system as a remote administration server, you must configure the server. See [section “Configuring the remote administration server”](#).
- configuring the ADM trap server
If you want to use your system as an ADM trap server, you must configure the server. See [section “Configuring the ADM trap server”](#).
- creating TNS entries
If you use the TNS you may need to create the TNS entries, see [section “Entering transport system applications in the TNS”](#).
If no or no current TNS entries are present for openFT V12 (because CMX was installed after openFT), then you can subsequently create or update these using a script, see [section “Entering transport system applications in the TNS”](#).

Please note that cluster configurations are only supported for TCP/IP. You are therefore recommended to work without CMX and TNS.

3.2.1 Checking the default settings

In the case of a new installation, openFT sets default values for the operating parameters and FTAC settings. These are chosen in such a way that they generally suffice for normal openFT operation. However, you should check whether these settings are suitable for your particular application and requirements. The special functions such as remote administration server, trace, traps, automatic deletion of log records etc. as well as the use of TNS and CMX are deactivated.

The default admission set is defined in such a way that unrestricted file transfer is possible. As FTAC administrator you should therefore modify the standard admission set to match the security needs of the computer (see also [section "Administering admission sets"](#)).

Operating parameter settings

Following a new installation (including the installation of openFT-CR and release of openFT-FTAM and openFT-FTP), you can use the `ftshwo` command to display the settings:

```
ftshwo
STARTED PROC-LIM CONN-LIM ADM-CLIM RQ-LIM MAX-RQ-LIFE TU-SIZE CCS-NAME
  YES      2      16      8      2000      30      65535      CP1252
PTN-CHK DYN-PART SEC-LEV FTAC-LOG FT-LOG FT-DIR-LOGADM-LOG USE TNS USE CMX
  STD      ON      B-P-ATTR  ALL      ALL      NONE      ALL      NO      NO
OPENFT-APPL FTAM-APPL FTP-PORT ADM-PORT ADM-CS
*STD      *STD      21      11000      NO
ACTIVE      ACTIVE      ACTIVE      ACTIVE
RSA-PROP RSA-MIN AES-MIN ENC-MAND
2048      0      NONE      NO
HOST-NAME IDENTIFICATION / LOCAL SYSTEM NAME
*NONE      mc011.mynet.local / $FJAM,MC011
FT-ADMIN FTAC-ADMIN
SYSTEM *STD
DEL-LOG ON AT RETPD RECOVERY ADM-TRAP-SERVER
  OFF DAILY 00:00 14 IN+OUT *NONE
TRAP: SS-STATE FT-STATE PART-STATE PART-UNREA RQ-STATE TRANS-SUCC TRANS-FAIL
CONS OFF OFF OFF OFF OFF OFF OFF OFF
ADM OFF OFF OFF OFF OFF OFF OFF OFF
FUNCT: SWITCH PARTNER-SELECTION REQUEST-SELECTION OPTIONS OPTIONS-LL
MONITOR OFF ALL ALL
TRACE OFF ALL ALL NONE OFF
```

For a detailed description of the individual values, see the manual "openFT (Unix and Windows systems) - Command Interface".

You should first of all check the following settings:

- Instance ID

This is preset to the name of the computer in the local network. If this is not sufficient to ensure unambiguous identification throughout the network, please change the value (`ftmodo -id`), see also [section "Unique Instance Identifications"](#).

- Local port numbers for the protocols (OPENFT-APPL, FTAM-APPL, FTP-PORT).

If you use other port numbers for addressing, e.g. for openFT, FTAM or FTP, then you should adapt these (`ftmodo` command, options `-openft`, `-ftam` and `-ftp`).

- Dynamic partners (DYN-PART)

Dynamic partners are permitted. If you want to lock this for security reasons then use `ftmodo -dp=f` to set the value to OFF.

You can also change the operating parameter settings in the openFT Explorer. To do this, open the *Administration* menu and choose the *Operating Parameters - Options* command.

FTAC settings

Following a new installation, all the values for the default admission set are set to 100. This means that the system is open for all users with a valid password, for all partners and for all actions. You should adapt the default admission set to meet the security needs of the system. You can do this using the `ftmoda` command, see manual "openFT (Unix and Windows systems) - Command Interface". Alternatively, you can use the openFT Explorer and adapt the *STD admission set in the *Admission Sets* object window.

3.2.2 Importing configuration data

You can import configuration data that you have created in another system, for example by means of an export:

- You import operating parameter settings by executing the export file in the command prompt.
- You import the FTAC environment by means of the *ftimpe* command or in the openFT Explorer by opening the *Administration* menu and choosing the *FTAC Environment - Import FTAC Environment* command.
- You import a partner list by executing the export file in the command prompt.
- You import the configuration for a remote administration server by means of the *ftimpc* command or in the openFT Explorer by opening the *Administration* menu and choosing the *Remote Administration - Import Configuration* command.

3.2.3 Activating and disabling the automatic startup of openFT

For every openFT instance that you create manually with *ftcrei*, the asynchronous openFT server is preset so that it is not automatically started when the system is booted.

You can activate and deactivate automatic startup of the asynchronous openFT server in the openFT Explorer by choosing *Administration - Operating Parameters - General* and toggling the *Start Asynchronous Server Automatically* option.

3.2.4 Creating the partner list from TNS

Due to the use of the partner list, openFT makes it possible to work without TNS provided that openFT communicates with partners via TCP/IP. Compared to TNS, the partner list has the advantage that you can use it to store not only all the necessary address information but also other properties such as, for example, a partner's security level.

If you want to switch to operation without TNS then you can use the tool *tns2ptn*. *tns2ptn* is used to create new partner list entries on the basis of TNS entries with the RFC1006 address format.

You must perform the following steps to insert TNS entries in the partner list:

1. Export the TNS entries to a file:

Call the program *TNS User interface*, open the *File* menu, select *Export TNS Entries* and enter an appropriate file name in the dialog box, e.g. *openft.tns*.

2. If necessary, clean up the export file (here *openft.tns*) by deleting the entries that do not relate to openFT, are no longer required or do not have the RFC1006 address format.

3. Call the tool *tns2ptn*:

```
openFT installation directory\bin\ftbin\tns2ptn.exe openft.tns > ft_list.bat
```

ft_list.bat is the name of the output file and can be selected freely. *ft_list.bat* contains an *ftaddptn* command with the associated address information for each partner.

If an entry cannot be converted then it is output at *stderr*.

4. Run the output file (here *ft_list.bat*) as FT administrator at command level.

Please note that the address information is taken over from TNS. Additional partner properties (security level, priority, tracing etc.) can subsequently be defined using the *ftmodptn* command or via the openFT Explorer.

4 Operating

- Optimizing the operating parameters
- Administering code tables
- Starting and stopping the asynchronous openFT server
- Setting operating modes (Windows systems)
 - Running the service under system rights
 - Running the service under user rights
- Setting the protection bit for newly created files (Unix systems)
- File access rights for newly created files (Windows systems)
- File access under user rights (Unix systems)
- Single-user mode (Unix systems)
 - Switching to single-user mode
 - Returning to multi-user mode
 - Note on the ftalarm function in single-user mode
- Switching the language interface
 - Switching the language interface on Unix systems
 - Switching the language interface on Windows systems
- Administering requests
- Administering partners
 - Exporting the partner list
- Security in FT operation
 - Authentication
 - Unique Instance Identifications
 - Creating and administering local RSA key pairs
 - Importing keys
 - Exporting public keys
 - Administering the keys of partner systems
 - Distributing the keys to partner systems
 - Extended authentication check
 - Encryption on data transfer

-
- Monitoring and controlling FT operation
 - openFT logging
 - SNMP interface of openFT (Windows)
 - Activities after installation
 - Starting the openFT subagent on Windows
 - SNMP management for openFT (Windows)
 - Monitoring with openFT
 - Configuring monitoring
 - Displaying monitoring data
 - Administering the FTAC environment
 - Administering admission sets
 - Administering admission profiles
 - Saving and migrating the FTAC environment
 - openFT instances and cluster operation
 - Save and restore configuration data
 - Characteristics of FTAM on Unix and Windows systems
 - Mapping file access rights
 - Outbound requests
 - Inbound requests
 - Mapping FTAM attributes to the real file system
 - Inbound mapping of FTAM attributes
 - Inbound mapping the document type
 - Access protection
 - Outbound mapping of the document type

4.1 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.

The following parameters are available for controlling the operation of openFT. You can specify these parameters by means of the *ftmodo* command:

- The maximum number of asynchronous requests that openFT should process simultaneously (connection limit).
- The maximum number of processes that are available for processing asynchronous requests (process limit).
- The upper limit for the length of blocks to be transferred.
Following the installation of openFT/openFT-FTAM, the maximum block length is set to 65535 characters.
- The maximum lifetime of transfer requests

You can view the current values of the parameters for an openFT instance with the *ftshwo* command.

You can also view and change the current operating parameters via the openFT Explorer. To do this, open the *General* tab and *Options* tab of the *Operating Parameters* window by selecting the appropriate menu item in the *Administration* menu. You will find a detailed description of each function in the online help.

Tips for performance control

When specifying the value for the process limit (PROC-LIM) and the connection limit (CONN-LIM), you must consider the following points:

i *Unix systems:*

You can only set the process limit to 1 or "Unlimited". If the value is "Unlimited" then the number of processes is determined by the connection limit (CONN-LIM) since each process handles only one connection.

- A low value for the process limit means that the requests are distributed across just a few processes and are therefore processed more slowly, but that on the other hand the performance of other applications on your computer is not significantly impacted.
- A high value for the process limit means that the requests are distributed over more processes and are therefore processed more quickly. On the other hand, increasing the process limit by too great an amount can cause the throughput to level off or even fall. In addition, the performance of other applications on your computer will be impacted to a greater extent.
- A low value for the connection limit means that only a few file transfers can run concurrently, and that connection requests from remote partners will be rejected more often because the limit is exceeded. The performance of other applications on your computer will not be degraded significantly.
- A high value for the connection limit means that a high volume of file transfer requests will be processed concurrently and will therefore be handled in a short period of time and connection requests from remote partners will generally be accepted. The performance of other applications on your computer will, however, possibly be degraded to a greater extent.

4.2 Administering code tables

A code table defines a character set (Coded Character Set, CCS for short) and the coding of these characters in the file. A CCS is assigned a name of up to 8 characters in length via which the CCS can be addressed.

As FT administrator, you can use the `ftmodo -ccs` command to set a standard CCS for openFT. In addition, you are still able to set your own 8-bit CCS.

The standard CCS is used for all FT requests. However, users can set a different CCS in the `ft-/ncopy` request and in the openFT Editor.

The following CCSs are supplied with openFT as standard:

Name of the CCS	Meaning
ISO88591 to ISO8859B and ISO8859D to ISO8859G	for the ASCII tables ISO8859-1 to ISO8859-11 and ISO8859-13 to ISO8859-16
ISO646	for the international 7-bit ASCII table
ISO646DE	for the German 7-bit ASCII reference version
EDF041 to EDF04A, EDF04D and EDF04F	for the EBCDIC tables DF04-1 to DF04-10, DF04-13 and DF04-15
EDF03IRV	for the international 7-bit EBCDIC table defined by FSC
EDF03DRV	for the German 7-bit EBCDIC table defined by FSC
UTF16	for Unicode with UTF-16 coding (platform-specific endian)
UTF8	for Unicode with UTF-8 coding
UTFE	for Unicode with the UTF-E coding
UTF16LE	for Unicode with UTF-16 coding (little-endian)
UTF16BE	for Unicode with UTF-16 coding (big-endian)
UTFEIBM	for Unicode with the UTF-EBCDIC coding defined by IBM
IBM037	for the US/Canada EBCDIC character set defined by IBM
IBM273	for the German/Austria EBCDIC character set defined by IBM
IBM500	for the International EBCDIC character set defined by IBM
IBM1047	for the OpenExtensions EBCDIC character set defined by IBM

CP437	for the English (USA) OEM character set defined by Microsoft
CP720	for the Arabic OEM character set character set defined by Microsoft
CP737	for the Greek OEM character set defined by Microsoft
CP775	for the Lettish OEM character set defined by Microsoft
CP850	for the Western Europe OEM character set defined by Microsoft
CP852	for the Polish OEM character set defined by Microsoft
CP855	for the Serbian OEM character set defined by Microsoft
CP857	for the Turkish OEM character set defined by Microsoft
CP858	for the OEM character set CP850 with the Euro symbol defined by Microsoft
CP862	for the Hebrew OEM character set defined by Microsoft
CP866	for the Cyrillic OEM character set defined by Microsoft
CP874	for the Thai Windows character set defined by Microsoft
CP1250	for the Central Europe Windows character set defined by Microsoft
CP1251	for the Cyrillic Windows character set defined by Microsoft
CP1252	for the Western Europe Windows character set with the Euro symbol defined by Microsoft
CP1253	for the Greek Windows character set defined by Microsoft
CP1254	for the Turkish Windows character set defined by Microsoft
CP1255	for the Hebrew Windows character set defined by Microsoft
CP1256	for the Arabic Windows character set defined by Microsoft
CP1257	for the Baltic Windows character set defined by Microsoft
CP1258	for the Vietnamese Windows character set defined by Microsoft

Creating a user-defined CCS

If you are an openFT administrator, you can create your own CCS (Coded Character Set). To do this, you must create a text file which is stored in the `sysccs` subfolder of the openFT instance. The CCS name corresponds to the name of this file.

The text file must have the following structure:

- The first line starts with a '#'.
The second character is a blank. The remainder of the line contains a comment which characterizes the code contained.
- The second line contains an alphabetic character which can at present only have the value 'S'. 'S' stands for single-byte code, i.e. a character is always 1 byte in length.
- The third line contains three numbers.
The first number is a 4-digit hexadecimal number. This defines the substitution character to be used if a Unicode character cannot be mapped to the code.
The second number is currently always '0'.
The third number is a decimal number which defines the number of code pages that follow. It currently always has the value '1'.
- The following lines define the code pages and have the following structure:
 - The first of these lines contains the number of the code page in the form of a twodigit hexadecimal number.
 - All the subsequent lines contain the mapping of the characters for the codes to be defined to UTF-16 in the form of a 4-digit hexadecimal number. The values are arranged in 16 lines, each of which contains 16 4-digit hexadecimal numbers with no spaces.

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

```
# Encoding file: iso8859-15, single-byte
S
003F 0 1
00
0000000100020003000400050006000700080009000A000B000C000D000E000F
0010001100120013001400150016001700180019001A001B001C001D001E001F
0020002100220023002400250026002700280029002A002B002C002D002E002F
0030003100320033003400350036003700380039003A003B003C003D003E003F
0040004100420043004400450046004700480049004A004B004C004D004E004F
0050005100520053005400550056005700580059005A005B005C005D005E005F
0060006100620063006400650066006700680069006A006B006C006D006E006F
0070007100720073007400750076007700780079007A007B007C007D007E007F
0080008100820083008400850086008700880089008A008B008C008D008E008F
0090009100920093009400950096009700980099009A009B009C009D009E009F
00A000A100A200A320AC00A5016000A7016100A900AA00AB00AC00AD00AE00AF
00B000B100B200B3017D00B500B600B7017E00B900BA00BB01520153017800BF
00C000C100C200C300C400C500C600C700C800C900CA00CB00CC00CD00CE00CF
00D000D100D200D300D400D500D600D700D800D900DA00DB00DC00DD00DE00DF
00E000E100E200E300E400E500E600E700E800E900EA00EB00EC00ED00EE00EF
00F000F100F200F300F400F500F600F700F800F900FA00FB00FC00FD00FE00FF
```

4.3 Starting and stopping the asynchronous openFT server

By default, openFT is started automatically at system startup (under Windows systems as service).

The asynchronous openFT server executes asynchronous outbound requests, processes inbound requests, and (only on Windows systems) enables the use of admission profiles. If the asynchronous openFT server is not started, these functions cannot be used.

Starting and stopping the asynchronous openFT server manually

You can start and stop the asynchronous openFT server manually via the *ftstart* and *ftstop* commands or via the openFT Explorer with the *Administration/Start Asynchronous Server* or *Administration/Stop Asynchronous Server* functions.

Modifying the settings for automatic startup

- Unix systems

Automatic startup is preset in the startup file. If openFT is not to be started automatically, the relevant command line must be commented out from the startup file. See [section “Disabling the automatic startup of openFT”](#)).

Note: On Solaris systems, automatic startup is performed via SMF.

- Windows systems

You can use the openFT Explorer (*Administration/Operating Parameters...*, *Start Asynchronous Server Automatically* option) to specify whether the asynchronous openFT server should also be started automatically when the openFT service is started. Note that by default, the option for automatically starting the asynchronous openFT server is only activated for the *std* instance.

4.4 Setting operating modes (Windows systems)

The openFT service can run under user rights or system rights. By default, the service is started under system rights.

4.4.1 Running the service under system rights

This operating mode is the default setting and is recommended if more than one user is working on one system and true multi-user operation is required, e.g. on a central Windows server. It is recommended to retain this default setting.

Notes

- The service is automatically started up when the operating system is started up unless this has been explicitly deactivated via the administration facilities.
- The access to the file system and to the files in the network is performed exclusively with the rights of the user involved, i.e.
 - for inbound requests (requests with initiative in the partner system):
the owner of the FTAC profile or the initiator of the request who identified himself or herself with *user id*, *password*;
 - for outbound request:
the user who submitted the request in the local system).

For this to be possible, the service must change to the identity of the user for certain actions. To this end, the service requires the login password of the user involved which must be made known to the service in openFT Explorer via *Administration - User password...* or by means of the *ftsetpwd* command. The password must be stored in the following cases:

- for asynchronous access to NTFS files or UNC names
- if you want to use FTAC profiles for inbound requests
- for local and remote preprocessing, postprocessing and follow-up processing
- for access to files via UNC names
- Relative path names for inbound requests refer to the user-specific home directory. This can be defined by the system or domain administrator in the user management of Windows.
- The home directory specified for the corresponding user in the User Manager is used as the home directory. If no such directory is specified, openFT creates the directory. For Windows 10 for example, this directory is located in *user/userID.hostname* or *userID.domain*.

4.4.2 Running the service under user rights

When the openFT service is used under user rights, it is started under the rights of a predefined user. However, the user must possess administrator rights. When the service is run under user rights, you should note that the asynchronous openFT server only processes requests for the user under whose rights it was started.

This operating mode is reasonable

- if only one user is working on the system and true multi-user operation is therefore not required,
- or for automated procedures since these normally do not require multi-user functionality.

As Windows administrator, you set the user as follows (Windows 10):

1. Choose *Control Panel - Administrative Tools - Services* and select the service *openFT*.
2. Choose *Properties* in the context menu.
The *Log On* tab is displayed.

In the window which has now been opened, select *Log On As .. This Account* and specify the user name and the user password of the account under which the service is to be started.

If the service is to run under the system account again, select *Local System Account*.

Notes

- The service is automatically started up when the operating system is started up unless this has been explicitly deactivated via the administration facilities.
- Access to the file system with the rights of the user under whom the service is executing, i.e. the FT server, can also only access files which the user is also able to access. This enables access to files in the network via UNC names provided that the users themselves have access.
- As transfer admission, all the FTAC profiles of the user can be used with whose rights the service is running. Alternatively, a combination of *user id, password* of the user with whose rights the service is running will also be accepted as the transfer admission. Transfer admissions or user ids of other users will not be accepted.
- It is no longer necessary to store the password using the openFT Explorer (*Adminis-tration - User password*) in order to allow FTAC profiles to be used, NTFS files to be accessed for asynchronous outbound requests, and local (outbound) or remote (inbound) follow-up processing to be performed.

This menu item is still available at the openFT Explorer so that you can change to a different operating mode (service executes with system rights) at any time.

- The home directory specified for the corresponding user in the User Manager is used as the home directory. If no such directory is specified, openFT creates (for Windows 10, for example) the *userID.hostname* or *userID.domain* directory under the *user.* directory.

4.5 Setting the protection bit for newly created files (Unix systems)

You can set the protection bit value for new files created on reception to a value that restricts the file access rights for the owner, the group members and for other users.

You may modify the standard protection bit setting with the *umask* command. In order to activate the modification, you must restart the asynchronous openFT server after the change has been made.

To ensure that the protection bit value is properly set when openFT is started, you should activate the command line *umask 027* in the startup file for the standard instance *std*. This startup file is located under */var/openFT/std/etcinit/openFTinst*.

However, since as of openFT V12, SMF is always used in Solaris, you must use SMF commands to modify the protection bit setting.

In Solaris systems, you modify the umask setting as follows:

1. Shut down openFT using the *ftstop* command.
2. Change the umask setting (e.g. to 022) using the command:

```
svccfg -s openFT:std setenv -i OPENFTUMASK 022
```
3. Take over the settings using the following command:

```
svcadm refresh openFT:std
```
4. Start openFT using the *ftstart* command.
5. You can display the settings by entering the *svccprop* command (here for the standard instance):

```
svccprop -t -p method_context/environment openFT:std
```

Output:

```
method_context/environment astring OPENFTINSTANCE=std OPENFTUMASK=022
```

i As of version openFT V12.1B, the platform Solaris is only supported on request.

4.6 File access rights for newly created files (Windows systems)

You define the file access rights for newly created files via the openFT Explorer. Choose the *file Access Rights...* command in the *Administration* menu, and specify in the *File Access Rights* dialog box whether the received file is to be created with the standard access rights which apply to the directory or with full access for the user who created the file. This is relevant to NTFS-file systems only.

With Inbound traffic, either the owner of the profile currently being used or the user ID explicitly specified by the initiator is used.

You must restart the asynchronous openFT server after the change has been made.

4.7 File access under user rights (Unix systems)

As of openFT V12, file access on Unix systems is performed by default under user rights - unlike in earlier versions of openFT. As a result, openFT performs all admission checks and accesses relating to a user's files and directories under the rights of the relevant user, i.e. for admission checks and access attempts, openFT switches from the privileged *root* context to the user's rights context and then back again.

Switching to the user context has the advantage, for example in the case of mounted NFS directories, that the *root* ID no longer requires access to the user files since accesses are performed exclusively under the rights of the relevant user.

4.8 Single-user mode (Unix systems)

openFT runs in multi-user mode after a new installation and an update installation. As the FT administrator you can switch openFT to single-user mode and return to multi-user mode using the openFT command *ftsetmode*.

The *ftsetmode* command should not be invoked during running openFT operation because *ftsetmode* terminates the openFT activities of all active instances (e.g. openFT Explorer, *ftscript* jobs, *ftexec* and *ncopy* commands).

i If openFT is already installed and openFT is running in single-user mode, single-user mode is retained if openFT is installed again.

4.8.1 Switching to single-user mode

Invoke the following command under *root* (here for the user ID *user1*):

```
ftsetmode -s -ua=user1
```

openFT then runs completely under this user ID, the so called openFT ID. In this case all instances (the standard instance and all other active instances) are allocated to the openFT ID. The following applies:

- Other IDs than the openFT ID have no access to functions of the appropriate openFT instance (exceptions: *ftcrei*, *ftdeli*, *ftmodi*). The invoking of openFT commands from unauthorized IDs is rejected.
- Access to the openFT instance is only possible on the inbound side via FTAC profiles, which belong to the openFT ID and in which no explicit user password is specified.
- Admission sets and admission profiles for external IDs are invalid, but continue to exist.
- Accesses that specify user ID and user password are not possible.

Administration rights in single-user mode

After switching to single-user mode, the administration rights (FT, FTAC and ADM administrator) are assigned as follows :

- The openFT ID is FT administrator with the following exception: Creating, deleting and modifying openFT instances is not allowed for the openFT ID.
- The openFT ID is FTAC administrator, i.e. existing profiles for other user IDs can be deleted or modified if the modification does not refer to the user ID. In the admission set of the openFT ID the limit values for administrator and user are changed simultaneously when *ftmoda* is invoked.
- If the openFT ID was ADM administrator before switching, it remains ADM administrator.
- If an other user ID was ADM administrator before switching, the right is returned, i.e. the ADM right is no longer assigned to any user ID. The openFT ID then can assign this right to itself (*ftmoda -admpriv=y*) because it is FTAC administrator. By this, the openFT instance is able to operate as remote administration server also in single-user mode.

openFT instances in single-user mode

The following applies in single-user mode:

- The instance commands *ftrei* (creating and reactivating), *ftdeli* (deactivating) and *ftmodi* (modifying) can be invoked only under *root*. This means in detail:
 - When creating or modifying an instance always a user ID must be specified. Thereby *root* can create instances for any user ID oder assign existing instances to any user ID.
 - If an openFT instance is assigned to an other user ID (via *ftmodi*) then the new instance owner also is FTAC administrator of this instance. If the previous owner of the instance was ADM administrator then the ADM right is passed to the new owner.
- The instance commands *ftseti* (setting) and *ftupdi* (updating) are only allowed for the owner of the instance.

Single-user mode under Solaris

i As of version openFT V12.1B, the platform Solaris is only supported on request.

In single-user mode, the openFT instances that are enabled in the SMF are started "transiently" for the appropriate openFT ID. "Transiently" means that the openFT instances in single-user mode are not monitored by the SMF.

In single-user mode, the openFT ID can start or stop the openFT instance using the commands *ftstart* and *ftstop*. The settings in the SMF are unchanged, see [section "Solaris SMF"](#).

4.8.2 Returning to multi-user mode

Invoke the following command under *root*:

```
ftsetmode -m
```

For each active instance including the standard instance the following applies:

- FT and FTAC administrator are assigned to *root*. If the openFT ID was ADM administrator in single-user mode then the ADM right is returned.
- Admission sets and admission profiles for other user IDs are reactivated when returning to multi-user mode.
- After switching, openFT is restarted for the instances for which it had been started before the switch.

Reactivating a user-specific instance in multi-user mode

User-specific instance means that in single-user mode an instance has been created for an other user ID than *root* or has been assigned to an other user ID than *root*.

If a user-specific instance is reactivated in multi-user mode (*ftcrei*) then *root* will become the FT and FTAC administrator of this instance. If the previous owner was ADM administrator this right is passed to *root*.

4.8.3 Note on the *ftalarm* function in single-user mode

Deactivate the *ftalarm* function via *ftalarm -t* before switching between single-user mode and multi-user mode or assigning an openFT instance to an other user ID in single-user mode. The reason: Although the *ftalarm* function is still suspended in the *cron* function after the switch, it would however no longer be effective.

ftalarm -t can be invoked from any user ID. As a result, it is also possible to remove invalid *ftalarm* entries or those that have become invalid from the *cron* function.

ftalarm -i can be used to check whether the *ftalarm* function is activated for the invoking user and the set openFT instance.

4.9 Switching the language interface

On Unix and Windows systems, you can switch the language interface between English and German.

4.9.1 Switching the language interface on Unix systems

During installation on Solaris, Linux and AIX systems, the *LANG* environment variable of the administrator performing the installation is evaluated and the associated value set as the default for the language interface. On HP-UX systems, English is set by default.

This value can be changed as follows:

- The openFT administrator can change the default setting with the *ftlang* tool. Only the setting specified via the *ftlang* tool is relevant for the output of the man pages on the platforms Solaris, AIX and HP-UX. On Linux systems, the German and English man pages are installed, i.e. users see the man pages in the language set for their login sessions (dependent on the *LANG* variable).
- Each user can change his or her own language setting using the *OPENFTLANG* environment variable. The user must enter the first two letters of the language setting in the *LANG* variable (*de* or *en*) and then export the environment variable.

Example

```
OPENFTLANG=de; export OPENFTLANG corresponds to (for example):  
LANG=De_DE.88591,De_DE.646, etc.
```

or

```
OPENFTLANG=en; export OPENFTLANG corresponds to (for example):  
LANG=En_US.ASCII,En_US.88591, etc.
```

The following table shows the effects of setting (or not setting) the *OPENFTLANG* and *LANG* variables:

OPENFTLANG	LANG	Result
Not set or empty	Not set or empty	Default setting
Not set or empty	Invalid value	Default setting
Not set or empty	Valid language (German or English)	Language set in <i>LANG</i>
Invalid value or a language that is not installed	Not evaluated	Default setting
Valid value (de or en)	Not evaluated	Language set in <i>OPENFTLANG</i>

The changed language setting takes effect as soon as a program such as the openFT Explorer, the openFT Editor or the shell is called again. If a program was active before the change, you must first close it and then restart it.

4.9.2 Switching the language interface on Windows systems

The default language for openFT is set on installation:

- With operator-guided installation: If *German* or *English* is set as the locale in the operating system, this language is taken as the default language for openFT during installation. In the case of all other system languages, you are asked whether *German* or *English* is to be preset as the default language for openFT.
- With unattended installation: openFT is installed with the *German* language interface if you specify the `TRANSFORMS=:de` parameter for the Windows Installer. Otherwise, openFT is installed with the *English* language interface.

This language setting can be changed as follows:

- The user can change the locale using the Control Panel (e.g. *Region and Language* topic under Windows 10). This change then also applies to other programs.
- Using the environment variable OPENFTLANG, each user can modify their own language setting.
 - > call up the Windows Control Panel
 - > Choose *System and Security - System - Advanced System Settings*.
 - > In the dialog box that then opens, click *Environment Variables*.
 - > In the user variables, set the OPENFTLANG variable to *de* or *en*.

The following table shows how settings (or lack of settings) for OPENFTLANG and the locale set in the control panel work:

OPENFTLANG	Language setting	Result
Not set, empty or invalid value	Unsupported language	The language set on installation
Not set, empty or invalid value	Supported language (<i>German</i> or <i>English</i>)	Language set with language setting
Valid value (<i>de</i> or <i>en</i>)	Not evaluated	Language set with OPENFTLANG

The changed language setting takes effect as soon as a program such as the openFT Explorer, the openFT Editor or the Windows command prompt is called again. If a program was active before the change, you must first close it and then restart it.

4.10 Administering requests

The request queue stores all asynchronous outbound requests, and all inbound requests.

As the FT administrator, you can use the following commands to administer the request queue:

- *ftshwr*: obtain information about all asynchronous requests on your system that are not yet completed. This includes the right to query information about all requests of all users.
- *ftmodr*: modify the processing order of all requests on your system, including those of other users.
- *ftcanr*: cancel asynchronous requests on your system, including those of other users.

You can also administer requests in the openFT Explorer by clicking on the *Request Queue* object directory. There you can cancel asynchronous requests, update the request queue, change the priority of requests or move requests to the beginning or end of the queue

You will find detailed descriptions of the functions in the online help of the openFT Explorer.

4.11 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 "Administering 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.

For special cases, the **Transport Name Service** (TNS) is provided, see [section "Administering partners"](#).

Following a new installation, the partner list is empty. Consequently, you should create the partner list immediately after installation and, in particular, enter frequently used partners in this list.

You can use the following commands to administer the partner list:

- *ftaddptn*: Enter new partner in the partner list and define the properties of the partner.
- *ftmodptn*: Modify the properties of a partner in the partner list.
- *ftremptn*: Remove a partner from the partner list.
- *ftshwptn*: Display the properties of partners in the partner list and export the partner list.
- *ftmodo*: Define the global FTAC security level (*-s/* option), activating/deactivating dynamic partners (*-dp* option).

The components of the partner addresses is explained in the manual "openFT (Unix and Windows systems) - Command Interface".

Alternatively you use the openFT Explorer:

- You enter a new partner in the partner list via the menu command *File - New - Partner List Entry ...*
Alternatively: In the object hierarchy, click *Administration* and choose *New Partner List Entry...* from the *Partner List* context menu.
- Using context menu commands in the *Partner List* object window you can enter a new partner, delete a partner and change the attributes of a partner.
- You define global partner properties via the *Administration* menu, *Operating parameters* command, *Options* tab.

i The openFT Explorer also contains the object directory *Partners* in which the relevant user can set up his or her preferred connection partner.

Further details can be found in the online help.

Registered dynamic partners

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

```
ftaddptn -pa=address -tr=n
```

I.e. you assign one or more attributes that are different from the corresponding default values (in this example *-tr=*n**, i.e. activate trace). Please note that *s/=* without parameters (default setting for the *ftaddptn* command) is a differently set attribute but not *-s/=p* (default setting for free dynamic partners).

If you want to delete dynamic partners from the partner list you have to reset all attributes of the partner to default values using the *ftmodptn* command.

Transport Name Service

For details, see [section “Entering transport system applications in the TNS”](#).

To use the TNS you must make sure that the following prerequisites are satisfied.

- CMX must be installed because CMX makes both the TNS and functions for accessing the TNS available.
- You must explicitly activate the function in the operating parameters. To do this, you either enter the *ftmodo - cmx=y -tns=y* command or activate both functions via the openFT Explorer (operating parameter options *Use TNS* and *Use CMX*).

On Windows systems TNS can be approached in the openFT Explorer via *Administration - Partner TNS Addresses*, see [section “Administering partners”](#).

4.11.1 Exporting the partner list

You can use the *ftshwptn* command to export the partner list entries to a file, for example in order to back up the entries or use them in other systems. On export, the entries are converted into the corresponding commands (*ftmodptn*) which you simply need to read in.

In *ftshwptn* you also specify the platform for which the commands are to be generated.

Examples (Unix systems)

- To back up the partner list in a format for Unix systems in the file *ftpartner.sav*:

```
ftshwptn -px > ftpartner.sav
```

You can re-import the partner list by calling the file as a procedure file, e.g. with

```
sh ftpartner.sav
```

- To export the partner list in BS2000 format to the file *ftpartner.bs2*:

```
ftshwptn -p2 > ftpartner.bs2
```

Examples (Windows systems)

- To back up the partner list in a format for Windows systems in the file *ftpartner.bat*:

```
ftshwptn -pw > ftpartner.bat
```

You can re-import the partner list by calling the file as a batch file, e.g. with

```
scmd /c ftpartner.bat
```

- To export the partner list in BS2000 format to the file *ftpartner.bs2*:

```
ftshwptn -p2 > ftpartner.bs2
```

4.12 Security in FT operation

A higher level of security in file transfer is offered by the following functions:

- Authentication
- Extended authentication check
- Encryption on data transfer

Note

- In addition, an integrity check can be required in the transfer request (*ft/ncopy -di*) if encryption is not used.
- openFT is able to exchange encrypted outbound file contents with a Secure FTP server if openFT-CR is installed on the openFT side and the FTP server supports the TLS (Transport Layer Security) protocol. The Secure FTP server makes its key and the certificate available to the openFT instance for encryption purposes. No mutual authentication is carried out.

4.12.1 Authentication

When authentication is used the following topics are particularly important:

- Unique Instance Identifications
- Creating and administering local RSA key pairs
- Importing keys
- Administering the keys of partner systems
- Distributing the keys to partner systems

4.12.1.1 Unique 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 using the *-id* option of the *ftmodo* command. In the openFT Explorer, use the *Administration* menu, *Operating Parameters* command, *General* tab, *Identification* parameter.

Partner instance IDs

Instance IDs of partner systems should, from your local system's point of view, correspond to the partner address, by which the partner system is known in the openFT. Instance IDs of partner systems should, from your local system's perspective, correspond to the partner address by which the partner system is known to openFT. If this is not the case, you must enter the partner in the partner list and explicitly specify its instance ID.

Note the following:

- If you do not specify the instance ID when entering the partner in the partner list, the partner address is set as the default with openFT and ADM partners (without port number and/or transport selector if these were specified with the partner address). This means that the instance ID of the partner must then match the specified partner address (without port number/T selector).
- If your partner system is still a version of openFT equal to or older than V8.0, authentication is not supported. In this event, you should specify *%.<processor>.<entity>* (with the processor name and station name of the partner) as a dummy ID when entering the partner in the partner list, so that incoming requests from this partner can be assigned to this entry.

Alternatively, it is possible to resolve the name using a DNS or to make an entry in the *hosts* file (*/etc/hosts* on Unix systems) or in the TNS. When TNS is used the global name must correspond to the instance ID of the partner.

4.12.1.2 Creating and administering local RSA key pairs

An RSA key pair set is created during new installation of openFT and consists of private and public keys of suitable length.

You can use the following commands to generate and manage local RSA keys:

- *ftcrek* creates a RSA key pair set for the local openFT instance.
- *ftshwk* outputs the properties of all the keys in the local system.
- *ftupdk* updates public keys.
- *ftdelk* deletes local key pairs.
- *ftmodk* modifies RSA keys.
- *ftimpk* imports RSA keys.

You can also create and administer RSA key pair sets using the openFT Explorer. To do this, choose the relevant command from *Administration - Key Management*.

Key pair attributes

An RSA key pair set in the Unix or Windows system currently consists of five key pairs with a lengths of 768, 1024, 2048, 3072 and 4096 bits. Up to and including openFT version 12.1C10, an RSA key pair set in the Unix or Windows system consisted of only three key pairs with a lengths of 768, 1024 and 2048. Private keys are internally administered by openFT, public keys are stored in the *config* directory of the instance file tree of the openFT instance see [section “Installation of openFT” \(Unix systems\)](#) or [section “Installation of openFT” \(Windows systems\)](#) under the following name:

```
syspkf.r<key reference>.l<key length>
```

The key reference is a numerical designator for the version of the key pair. The public key files are text files that are created using the character code of the respective operating system, i.e. by default:

- Unix systems: ISO8859-1
- Windows systems: CP1252

Storing comments

In the *syspkf.comment* file in the *config* directory of the instance file tree, you can store comments, which are written in the first lines of the public key files when a key pair set is created. The *syspkf.comment* file is a text file that you can edit. The comments could, for example, contain the contact information of the FT administrator on duty, the computer name, or similar information that is important for partners. The lines in the file *syspkf.comment* can only be a maximum of 78 characters long. Using the command *ftupdk*, you can also import subsequent comments from this file into existing public key files.

Updating and replacing keys

If a public key file were accidentally deleted, you could re-create the public key files of the existing key pair set using *ftupdk*.

If you want to replace a key pair set with a completely new one, you can create a new key pair set using *ftcrek*. You will recognize the most up-to-date, public key 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, however, should be temporary, until you have made the most up-to-date public key available to all partner systems. Thereafter, you can delete key pair sets that are no longer needed using *ftdelk*. Deleted key pair sets can not be restored using *ftupdk*.

4.12.1.3 Importing keys

You can import the following keys using the *ftimpk* command or in the openFT Explorer (*Administration - Key Management*):

- 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 in the *config* directory in the instance file tree, see [section “Creating and administering local RSA key pairs”](#). This key can be used in the same way as a key generated with *ftcrek* 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* directory, see [section “Administering the keys of partner systems”](#).

Every imported key pair contains a unique reference number. RSA keys with the supported key lengths are imported (768, 1024, 2048, 3072 and 4096 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.

4.12.1.4 Exporting public keys

You can use the openFT explorer to export public keys of the local openFT instance and provide them partners with the keys required to authenticate their own instance (menu *Administration - Key Management*). Please refer to the online help for Details.

4.12.1.5 Administering the keys of partner systems

Public keys are stored in the *config* directory of the instance file tree of the local openFT instance see [section “Installation of openFT”](#) (Unix systems) or [section “Installation of openFT”](#) (Windows systems).

The public keys of the partner systems have to be stored as files in the directory *syskey* of the instance file tree of the local openFT instance, see [section “Installation of openFT”](#) (Unix systems) or [section “Installation of openFT”](#) (Windows systems). The instance ID of the partner system must be selected as the file name.

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

- You can call the *ftimpk* command and enter the name of the key file. openFT saves the key in the *syskey* directory and uses the partner’s instance identification in the correct notation (lowercase) as the file name.
- You can use the methods made available by the operating system to save the key file in the *syskey* directory under the partner instance ID name.
In Unix systems the file name must not contain any uppercase characters. If the ID contains any uppercase characters, they must be converted to lowercase characters.

If an updated, public key is made available by the partner instance, the old key file must be overwritten at that time.

You can use the *ftshwk* command to display the keys of partner systems (option *-pn*) and filter these on expiration date (option *-exp*).

For Secure FTP, some special features apply, see [section “Monitoring and controlling FT operation”](#).

Modifying the key attributes of partner systems

You can use the *ftmodk* command to modify the key attributes of partner systems by specifying an expiration date (*-exp* option) or modifying the authentication level (1 or 2, *-al* option):

- 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.

4.12.1.6 Distributing the keys to partner systems

Distribution of public key files to your partner systems should take place using reliable means, for example by

- distributing them via cryptographically secure by e-mail
- distributing them on a CD (by courier or by registered mail).
- distributing them via a central, openFT file server, whose public key is in the partners' possession.

You must ensure that your public key files are re-coded (e.g. by transferring them as text files via openFT), if you transfer them to a partner with a different operating system (e.g. BS2000 system).

The public key file of your local openFT instance is stored in the partner system in the following location:

- For partners using openFT (BS2000) as type D, PLAM elements in the library *SYSKEY* on the configuration user ID of the partner instance. The partner name allocated to your openFT instance in the remote partner list *SYSPTF* must be selected as the element name.
- For partners using openFT (Unix systems) in the directory *syskey* of the instance file tree. In the case of the standard instance the path name is */var/openFT/std/syskey*. The instance ID of your local openFT instance must be selected as the file name. The file name must not contain any uppercase characters. If the instance ID contains any uppercase characters, they must be converted to lowercase characters in the file name.
- For partners using openFT (Windows) in the directory *syskey* of the instance file tree. In the case of the standard instance the path name under Windows 10 is
%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 using openFT (z/OS) as a PO element in the library *admuser.instance.SYSKEY*, where *instance* is the name of the instance. The partner name allocated to your openFT instance in the remote partner list *SYSPTF* must be selected as the element name.

4.12.2 Extended authentication check

Extended authentication check means:

- for openFT partners that openFT is checking not only the processor name, but also the transport address.
- for FTAM partners that the Calling Application Entity Title and the transport address are checked. This prerequisites that the partner identifies itself with a Calling Application Entity Title which does not correspond to the nil AP Title.

FTAM partners are identified via the transport address by default.

The extended sender checking can be controlled as follows:

- globally for openFT and FTAM partners using *ftmodo* with option *-ptc* (see table below) or via the openFT Explorer using *Operating Parameters - General, Partner Check*

ftmodo	openFT partners	FTAM partners
<i>-ptc=i</i>	Instance identification	transport address
<i>-ptc=a</i>	Instance identification + transport address	transport address
<i>-ptc=t</i>	Instance identification	AET
<i>-ptc=b</i>	Instance identification + transport address	AET + transport address

For openFT partners, the global setting applies to all partners for which partner checks are set as default (FTOPT output in *ftshwptn*) and which do not operate with authentication.

- partner specifically for openFT partners, using *ftaddptn / fmodptn -ptc=a* (on) or *-ptc=i* (off) or via the openFT Explorer using the *Partner Check* section in the *Partner List Entry* dialog

In the case of dynamic partners and FTP partners, the sender check operates exclusively via the transport address. Consequently the "extended sender verification" attribute is ineffective and is also not displayed.

4.12.3 Encryption on data transfer

Prerequisites for encrypting user data:

- For openFT versions <= V12.1B:
openFT-CR must be installed both locally and on the partner system. For legal reasons, openFT-CR is not available in all countries.
- An RSA key pair set must exist in the local system and encryption must not be deactivated (encryption is deactivated by specifying *ftmodo -kl=0*).

You can check this using the *ftshwo* command. The output parameter RSA-PROP displays the length of the currently used RSA key in bits (0, 768, 1024, 2048, 3072 or 4096). 0 means that encryption is deactivated.

You can set the length required for the RSA key and the minimum RSA key length via the operating parameters. In addition, you can set a minimum AES key length.

To do this, use the following options:

- *-kl* (desired RSA key length) and *-klmin* (minimum RSA key length) in the *ftmodo* command or the openFT Explorer (*Administration* menu, *Operating Parameters* command). The default values after a new installation are 2048 (*-kl*) and 0 (*-klmin*).
- *-aesmin* (minimum AES key length) in the *ftmodo* command or the openFT Explorer (*Administration* menu, *Operating Parameters* command). The default value after a new installation is *-aesmin=* (no minimum AES key length set).

For further details on local keys, see [section “Creating and administering local RSA key pairs”](#).

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, specify the *ftmodo* command with the option *-c*. Alternatively, in the openFT Explorer: *Administration* menu - *Operating Parameters*, *General* tab, *Encryption of User Data*.

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. FTAM requests and inbound FTP requests are rejected because encryption is not supported. File management continues to be performed without encryption independently of the settings.

4.13 Monitoring and controlling FT operation

You can monitor and control FT operation by:

- openFT logging
- SNMP interface of openFT (Windows)
- Monitoring with openFT

4.13.1 openFT logging

The log records are stored in the file *syslog.Lyymmdd.Lhhmmss*. This file is located in the *log* directory of the relevant openFT instance. *yymmdd* is the date (year, month, day) and *hhmmss* the time (hour, minute, second for GMT) at which the file was created.

Displaying log records

You can use the *ftshwl* command to view all log records in the system. Using the polling options *-po* and *-pnr* provided by *ftshwl*, you can also repeat the output of new log records at regular intervals.

The output of a log record contains an RC column which indicates either 0000 (positive acknowledgment) or the cause of rejection or abort of the request by means of a 4-digit reason code. You can use the *fthelp* command to determine the meaning of the reason codes.

You can also view log records in the openFT Explorer. To do this, click on *Logging* under *Administration* in the navigation area.

Switching log file and administering offline logging

You can change the log file with the *ftmodo -lf=c* command

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 as often as you wish and therefore manage multiple offline log files.

You can display the names of all log files via *ftshwl -llf* (current log file and offline log files).

In the openFT Explorer, you can also see all the offline log files and associated log records (*Administration* subtree - *Logging - Offline Logging*).

Modifying log settings

You can modify the following log settings:

- The scope of logging (FT, FTAC and administration function as well as the logging for directory transfer). To do this, use the *ftmodo* command (options *-lt*, *-lc*, *-la* and *-ltd* (directory transfer)).

Following installation, full logging is set.

- The intervals for the automatic deletion of log records. To do this, use the *ftmodo* command (options *-ld*, *-lda*, *-ldd* and *-ldt*) or the *Logging* tab in the openFT Explorer under *Administration - Operating Parameters*.

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.

In the openFT Explorer use or the *Logging* tab under *Administration - Operating Parameters*.

Saving of log records in files and deleting log records

All log records may be deleted by the openFT administrator, the FTAC administrator and the ADM administrator. To do this, use the *ftdell* command or the openFT Explorer (Logging object window, context menu command *Delete Log Records*). Deleting log records causes the size of the log file to change since the storage space is immediately free upon deletion.

Saving log records

Redirect the output of *ftshwl* to a file as follows:

- If you want to back up current log records, call *ftshwl* without specifying *-lf*, *-tlf* or *-plf*. When you do this, select the log records that you want to back up. Then remove these log records from the current log file by calling *ftdell* with the appropriate selection criteria.
- If you want to back up offline log records, call *ftshwl -nb=@a* and specify *-lf*, *-tlf* or *-plf*. These options allow you to specify the offline log files. Next, delete the log file or files by calling *ftdell* with the option *-tlf*.

4.13.2 SNMP interface of openFT (Windows)

Prerequisites

On Windows systems, the Microsoft SNMP Server must be available. You have to install the SNMP sub agents for openFT explicitly, see [section “Installation of the SNMP subagent”](#).

i SNMP is no longer supported on Unix systems.

4.13.2.1 Activities after installation

After installation of openFT, the following activities are required.

1. If your system is not already being managed with SNMP, you will need to activate administration via SNMP.
You will need a community string with write authorization to manage openFT via the openFT subagent. If you only have read-only authorization (Windows systems), then only information can be output via SNMP. In this case you will not be able to change values (or perform starts or stops, see also [section “SNMP management for openFT\(Windows\)”](#)).
For further details please refer to [section “Installation of the SNMP subagent”](#).
2. Start the agent (see below)

i You will find a list of activities performed by the SNMP administrator in the documentation for the management station used.
Consult your SNMP documentation to obtain information on security mechanisms.

4.13.2.2 Starting the openFT subagent on Windows

The openFT subagent is registered with the SNMP service when the openFT SNMP function is installed. To start the openFT subagent, you must stop and restart the SNMP service once following installation. After this, the openFT subagent is started automatically each time the SNMP service is started.

i Note that SNMP can only work with one instance when clustered. The decisive factor is which instance is set up to start when the agent is started (see also [section “openFT instances and cluster operation”](#)).

4.13.2.3 SNMP management for openFT (Windows)

The openFT subagent is used to:

- obtain information about the status of asynchronous openFT server
- start and stop the asynchronous openFT server
- obtain information about system parameters
- modify system parameters
- create the new public key for encryption/authentication
- output statistical data
- to control the diagnosis

The MIB (Management Information Base) to openFT offers objects for the abovementioned management tasks. It is located in the file

- *openFT-installation-directory\snmp\openFTMIB.txt*

The objects for starting and stopping, encrypting the public key, modifying the system parameters and controlling the diagnose require write access.

Starting and stopping openFT

MIB definition

Object name/object identifier	Access	Meaning
ftStartandStop/1.3.6.1.4.1.231.2.18.1.1.0	read-write	openFT protocol

Input

Syntax	Integer	Meaning
start	1	the asynchronous openFT server is started
stop	2	the asynchronous openFT server is stopped

Output

Syntax	Integer	Meaning
on	3	the asynchronous openFT server is started
off	4	the asynchronous openFT server is stopped

Setting the values “start” or “stop” causes the openFT subagent to start or stop the asynchronous openFT server. Reading access supplies information about the current status of the FT system (“on” or “off”).

System parameters

MIB definition

Object name/object identifier	Access	Meaning	Command <i>ftmodo</i>
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	<i>-tu</i>
ftSysparMaxOSP/1.3.6.1.4.1.231.2.18.2.7.0	read-write	Max OSP ¹	<i>-cl</i>
ftSysparMaxISP/1.3.6.1.4.1.231.2.18.2.8.0	read-write	Max ISP ¹	<i>-cl</i>
ftSysparProcessorName/ 1.3.6.1.4.1.231.2.18.2.9.0	read-write	Processor Name	<i>-p</i>
ftSysparStationName/ 1.3.6.1.4.1.231.2.18.2.10.0	read-write	Station Name	<i>-l</i>
ftSysparCode/1.3.6.1.4.1.231.2.18.2.11.0	read-write	Code Table The following values are supported: iso8859-1 (1), iso8859-2 (2), iso8859-5 (5), iso8859-6 (6), iso8859-7 (7), iso8859-9 (9), undefined (255)	<i>-ccs</i>
ftSysparMaxInboundReqs/ 1.3.6.1.4.1.231.2.18.2.12.0	read-write	Max Inbound Requests	<i>-rql</i>
ftSysparMaxLifeTime/ 1.3.6.1.4.1.231.2.18.2.13.0	read-write	Max Life Time	<i>-rqt</i>

¹The distinction between *Max OSP* (maximum number of parallel outbound connections) and *Max ISP* (maximum number of parallel inbound connections) is no longer supported as of openFT V11. Both values correspond to the parameter *-cl* (connection limit) of the *ftmodo* command according to the following formula: $Max\ OSP = Max\ ISP = connection\ limit * 2/3$ (rounded to the nearest integer).

The explanation of the possible values in the description of the *ftmodo* command.

Statistical information

MIB definition

Object name/object identifier	Access	Meaning
ftStatSuspend/1.3.6.1.4.1.231.2.18.4.1.0	read-only	Requests in status SUSPEND
ftStatLocked/1.3.6.1.4.1.231.2.18.4.2.0	read-only	Requests in status LOCKED
ftStatWait/1.3.6.1.4.1.231.2.18.4.3.0	read-only	Requests in status WAIT
ftStatActive/1.3.6.1.4.1.231.2.18.4.4.0	read-only	Requests in status ACTIVE

ftStatCancelled/1.3.6.1.4.1.231.2.18.4.5.0	read-only	Requests in status CANCELLED
ftStatFinished/1.3.6.1.4.1.231.2.18.4.6.0	read-only	Requests in status FINISHED
ftStatHold/1.3.6.1.4.1.231.2.18.4.7.0	read-only	Requests in status HOLD
ftStatLocalReqs/1.3.6.1.4.1.231.2.18.4.8.0	read-only	local requests
ftStatRemoteReqs/1.3.6.1.4.1.231.2.18.4.9.0	read-only	remote requests

The individual states have the following meanings:

SUSPEND

The request was interrupted.

LOCKED

The request is temporarily excluded from processing.

This state may occur both for openFT and for FTAM partners.

With openFT partners, e.g. when a resource bottleneck is encountered or when external data media must be made available.

With FTAM partners, when one of the partners proposes a waiting period until the next start or recovery attempt via the FTAM protocol, and this period exceeds the delay normally permitted.

WAIT

The request is waiting.

ACTIVE

The request is currently being processed.

CANCELLED

The request was cancelled in the local system. However, the remote system is aware of its existence, e.g. because it was previously active. Therefore, the request cannot be removed from the request queue until a connection to the partner has been re-established.

FINISHED

This status arises for requests involving FTAM partners when the request has been either completed or cancelled, but the user has not yet been informed of the fact

HOLD

The start time specified when the request was issued has not been reached

Control of diagnostics

MIB definition

Object name/object identifier	Access	Meaning
ftDiagStatus/1.3.6.1.4.1.231.2.18.5.1.0	read-write	Diagnosis Management

Input

Syntax	Integer	Meaning
off	1	Diagnosis management is deactivated
on	18	Diagnosis management is activated

If the values are set to "on" or "off", the openFT subagent causes diagnostics management (tracing) to be started or stopped respectively. Read access provides information on the current status of diagnostics management (activated or deactivated).

Public key for encryption

MIB definition

Object name/object identifier	Access	Meaning
ftEncryptKey/1.3.6.1.4.1.231.2.18.3.1.0	write-only	Public key

Input

Syntax	Integer	Meaning
create-new-key	1	A new public key is created.

A detailed description on creating and managing public and private key can be found in [section "Creating and administering local RSA key pairs"](#).

4.13.3 Monitoring with openFT

You must be an FT administrator in order to configure monitoring. Each user can call up the monitoring data at any time provided that monitoring is activated and the asynchronous openFT server is started.

4.13.3.1 Configuring monitoring

You use the *ftmodo* command or the openFT Explorer (*Administration - Operating Parameters, Trace* tab) to configure monitoring. The following options are available:

- Activate and deactivate monitoring (*ftmodo -mon=*)
- Select monitoring by partner type (*ftmodo -monp=*)
- Select monitoring by request type (*ftmodo -monr=*)

Once the settings have been selected, they are retained until you explicitly change them. This means that they also remain unchanged after the computer has been rebooted.

You can check the current settings with the *ftshwo* command. The MONITOR row indicates whether monitoring is activated and shows any criteria used for selection.

4.13.3.2 Displaying monitoring data

The following options are available:

- Command *ftshwm*: Displaying local monitoring data.
- openFT Monitor: Displaying local or remote monitoring data

You start the openFT Monitor either using the openFT Explorer (*Extras* menu or the context menu of a partner entry) or using the *ftmonitor* command.

If you want to view the monitoring data of openFT instances on the other systems, you specify the partner and the transfer admission when you call the openFT Monitor. This is done implicitly in the openFT Explorer if you start the openFT Monitor from the context menu of an entry in the *Partner* object directory. In order to do this, you must activate the *Remote Command Execution* and *Administration Objects* options in the properties of this partner.

- Preprocessing: Displaying remote monitoring data

To do this, you define an admission profile by specifying a file name prefix with the keyword **FTMONITOR* as a preprocessing command, see [section “Displaying monitoring data”](#)

- Windows Performance Monitor: Displaying local monitoring data (only on Windows systems), see [section “Displaying monitoring data”](#).

Setting up an admission profile for displaying monitoring data

This example shows how you set up an admission profile for preprocessing on the remote system (1.) and how you can use it to perform output via the openFT Monitor (2.) and linebased output (3.).

1. Define an admission profile *monitor1* on the remote system *Partner1* that only permits the output of monitoring data. Assign *onlyftmonitor* as the transfer admission.

- Unix or Windows system:

```
ftcrep monitor1 onlyftmonitor -ff=tp -fnp="|*FTMONITOR "
```

- BS2000 system:

```
/CREATE-FT-PROFILE NAME=MONITOR1 -  
  ,TRANSFER-ADMISSION=ONLYFTMONITOR, -  
  ,FILE-NAME=*EXPANSION ('|*FTMONITOR ') -  
  ,FT-FUNCTION=( *TRANSFER-FILE, *FILE-PROCESSING)
```

- z/OS system:

```
FTCREPRF NAME=MONITOR1  
  .TRANSFER-ADMISSION=ONLYFTMONITOR -  
  ,FILE-NAME=*EXPANSION ('|*FTMONITOR ') -  
  ,FT-FUNCTION=( *TRANSFER-FILE, *FILE-PROCESSING)
```

i The asterisk (*) in **FTMONITOR* in the profile *monitor1* must be specified. It is furthermore recommended to enter a space after **FTMONITOR* in the profile itself, in order that subsequent options are automatically separated from the command.

2. You can specify the transfer admission of this profile in the *ftmonitor* command if you wish to view the openFT monitoring data from a remote system.

```
ftmonitor -po=10 Partner1 onlyftmonitor
```

In order to call the graphical openFT Monitor from the openFT Explorer, define a partner with this transfer admission in the *Partners* object directory.

3. Alternatively, you can use this FTAC profile to get the monitoring data in the form of linebased output and redirect it to a file for further processing using an *ft* or *ncopy* command. Note that at this point, only the interval can be set, but no monitoring data can be selected. Output is always in CSV format. The following command allows you to output the current monitoring values of *Partner1* at 10-second intervals:

```
ncopy Partner1!"-po=10" partner1_data onlyftmonitor
```

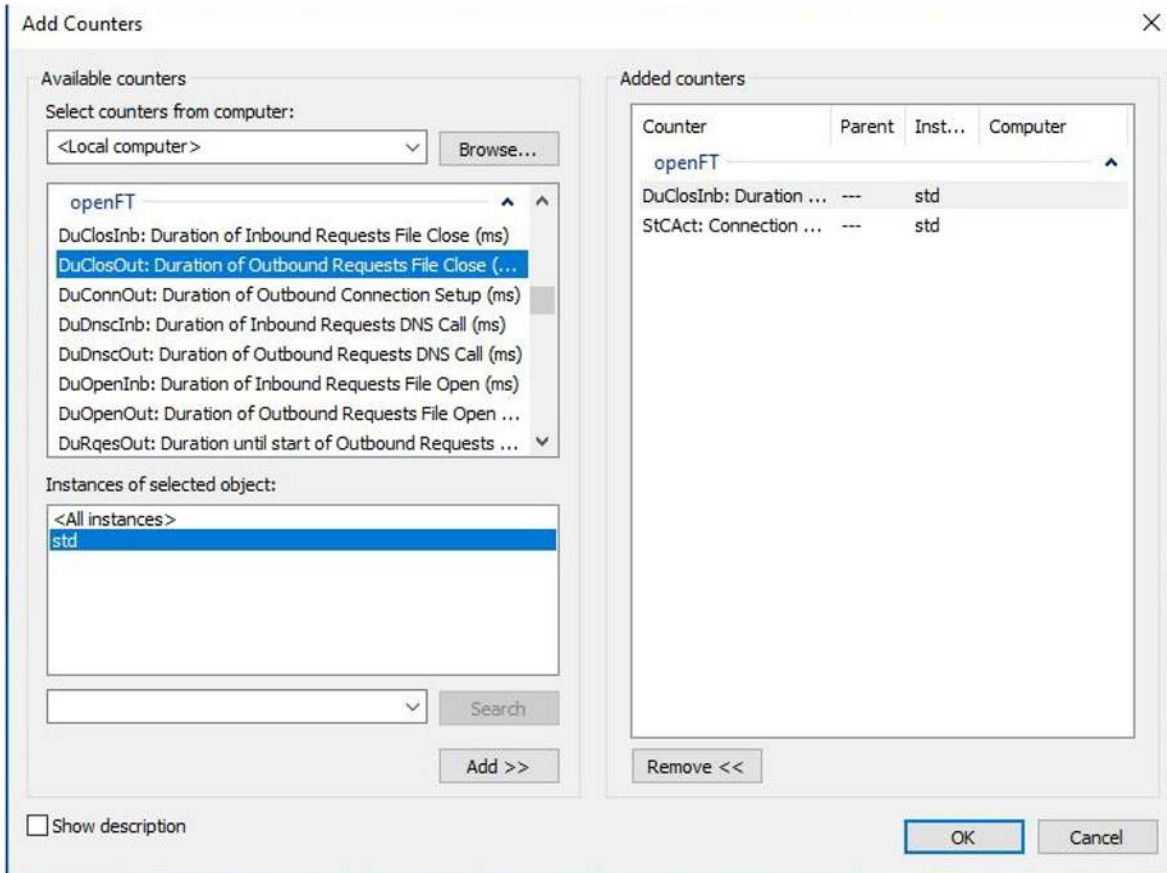
The monitoring data is output to the file *partner1_data*. The only parameter that you can specify within the quotes is *-po=polling interval*. If you wish to use the default polling interval of one second, enter a space between the quotes.

Displaying monitoring data via the Windows Performance Monitor (Windows systems)

To do this, first activate the required monitoring values via the Control Panel. Proceed as follows (the description applies to Windows 10):

- Start the Windows Control Panel and choose *System and Security - Administrative Tools - Performance Monitor* to open the *Performance Monitor* window.
- In the navigation bar, click on *Monitoring Tools - Performance Monitor* and click on the *+* button.
- In the *Add Counters* window, you will find a section entitled *openFT* under *Available counters*. Select the required openFT monitoring values and, if necessary, the openFT instance and click on *Add >>*. The monitoring values are listed on the right under *Added counters*.
- When you click OK, the monitoring values are recorded and displayed.

You can remove the openFT monitoring values from the Windows Performance Monitor again by selecting them and clicking on *Remove <<*.



4.14 Administering the FTAC environment

The term FTAC environment refers to the admission sets and admission profiles present on your system.

4.14.1 Administering admission sets

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

- You define the standard admission set using the command `ftmoda @s`.
Following an new installation of openFT, the standard admission set is set so that file transfer is possible without restriction. 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 admission profile.
- You can display admission sets of all users of the system using the `ftshwa` 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 `ftmoda`.
- You can specify the ADM administrator initially by setting the ADM privilege in the admission set of the ADM administrator (see [section “Defining the ADM administrator”](#)).

Alternatively you can use the openFT Explorer: In the navigation bar, click on *Admission sets* under *Administration* . All admission sets are listed in the *Admission Sets* object window. *STD is the standard admission set.

Using admission sets properly

With an openFT request (outbound and inbound), the admission specified in the admission set is compared with the FTAC security level of the partner concerned (see also [section “Exporting the partner list”](#)).

To protect your processor against attempted intrusion, you should set the inbound properties in the admission set as restrictively as possible for user IDs with administrator rights, i.e. at least prohibit inbound processing.

1. For secure operation, you should prevent all inbound admissions in the standard admission set, e.g. by using the command:

```
ftmoda @s -os=100 -or=100 -is=0 -ir=0 -if=0 -ip=0
```
2. For each user to whom inbound request may be processed, you, as FTAC administrator, should set all parameters of the corresponding admission set to 100.
3. Recommend all users to change their inbound values to 0. They may then use their profiles and the “ignore ... level” function to permit any desired access mode. Inbound requests for which the corresponding security level is 0 will then be allowed only via the FTAC transfer admission, but no longer via the login and password.

It is also possible,

- to assign partner-specific security levels, see [section “Exporting the partner list”](#)
- and for openFT partner to undergo a reliable identity check using cryptographic means, see [section “Authentication”](#).

The use of a file name prefix in the admission profile provides additional security. This prevents switching to a parent directory.

Important

If you have high security requirements, these actions are really only useful if no other network access options are available that allow the protection mechanisms to be circumvented. In particular, this means that TCP/IP services such as *ftp*, *ftps* must not be active.

4.14.2 Administering admission profiles

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

- *ftcrep*: create admission profile
- *ftdelp*: delete admission profile
- *ftmodp*: modify admission profile
- *ftshwp*: show admission profile

Alternatively you can administer admission profiles via the openFT Explorer: In the navigation bar, click on *Admission Profiles* under *Administration*. You will find a detailed description of the object windows in the online help.

As the FTAC administrator, you have the option of modifying foreign admission profiles:

- You can view them with the command *ftshwp*. 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.
- You can delete them with the command *ftdelp*. This function is necessary, for example, after deletion of a login name, since the profiles are not automatically deleted when a login name is deleted. You should contact the user before you delete profiles from active login names.
- You can privilege them with the command *ftmodp (-priv=y)*, or conversely revoke privileges (*-priv=n*).
- You can also modify them with *ftmodp*. Access to the admission profile will then be blocked until the owner of the profile acknowledges these modifications by resetting the transfer admission to “valid”, for example with *ftmodp ... -v=y*. If you also possess the FT administrator rights or specify explicitly the password in the profile, then the profiles are not locked.

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.

You can also delete admission profiles via the openFT Explorer by selecting the *Delete* command from the context menu.

4.14.3 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 admission profiles and restoring them on the new processor. Furthermore, you can also create backup copies of the FTAC environment on your processor by this method.

The following commands are available for migrating and saving the FTAC environment:

- *ftexpe*: output FTAC environment to file
- *ftimpe*: import FTAC environment from file
- *ftshwe*: show FTAC environment from export file

You can also save and import admission sets and admission profiles via the openFT Explorer using the *FTAC Environment* command in the *Administration* menu. You will find a detailed description to it in the online help.

You can also view saved admission sets and admission profiles via the openFT Explorer by selecting the *Open Export File* command in the context menu of the *Exported Admissions* object in the navigation bar.

Saving and importing admission sets and admission profiles

- When saving the FTAC environment you can select the admission sets and admission 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 the values of the currently valid admission set.

If you have FT administrator rights as the FTAC administrator, the admission profiles that you import will be immediately available with the status that was set on exporting the profile. If you do not have FT administrator rights, imported profiles will initially remain locked for all user IDs and must be unlocked (e.g via *ftmodp -v=y*).

4.15 openFT instances and cluster operation

With openFT, you can run several openFT instances at the same time on a single host. These instances allow you to switch to a different computer already running openFT so that you can continue to use the openFT functionality when the initial host fails. You will find examples on how to use openFT in a cluster of Unix or Windows systems in the appendix.

A requirement for this is that openFT uses only the TCP/IP transport system. Other transport systems are not supported in a cluster and must also not be configured in the TNS. As a result, you are recommended to work without TNS and CMX. If you work without CMX then you also automatically work without TNS. In a cluster, the same version of openFT must be running on all the computers.

For systems that do not have TCP/IP there is currently only the standard instance.

openFT commands that call preprocessing, postprocessing or follow-up processing run in the same instance as the request that initiated the pre-, post- or follow-up processing.

If you administer openFT via SNMP, then please note when switching to the cluster that SNMP can only work together with one instance.

The decisive factor is which instance is set when the agent is started see [section “openFT instances and cluster operation”](#).

Command for administering instances

As an openFT administrator you can create, modify and delete instances. You can also set up instances and obtain information on instances.

- Creating or activating an instance

Using the command *ftcrei*, you can create a new instance or re-activate (switch on) a deactivated instance.

When an instance is created, the operating parameters, the profile files, and on Unix systems the startup and shutdown files are initialized as during a new installation.

When an existing instance is activated on Unix systems, the existing instance file tree, with the operational resources of the instance, is linked to the directory */var/openFT*.

If you create a new instance and wish to continue using the standard instance *std*, You must assign the standard instance a separate address in order to avoid address clashes.

- Modifying an instance

You can assign a different Internet host name to an instance with the *ftmodi* command.

Please note:

If you assign the standard instance *std* a host name, local requests to the address 127.0.0.1 used for test purposes, for instance, are no longer possible.

- Deactivating an instance

You can deactivate an instance with the *ftdeli* command. Deactivating an instance in this manner only removes:

- the symbolic link in the local */var/openFT* directory (Unix systems)
- the instance from the openFT instance management (Windows systems)

The instance file tree is not changed.

- Setting up an instance

You can select the openFT instance you want to work with using the *ftseti* command.

The command sets the OPENFTINSTANCE environment variable to the name of the instance.

You can also set up the instance via the openFT Explorer. As soon as there is more than one instance, then a list appears in the tool bar of the openFT Explorer from which you select the instance.

i The list box is only displayed if the instance is already present when the openFT Explorer is started. If the instance is created after the start of the openFT Explorer then this must be restarted.

- Outputting information on instances

You can query information on the instances using the *ftshwi* command.

- Updating an instance file tree

Using the *ftupdi* command, you can modify the instance file tree of an older version of openFT for use in the current version. That is only necessary for instances that were not active at the time of an update installation.

- i**
- If you work with more than one instance on a Unix system, then in this case a separate *ftalarm* call is required for each instance.
 - Please consider also the [section "Note on the ftalarm function in single-usermode"](#).
 - You will find detailed descriptions of the *ftcrei*, *ftmodi*, *ftupdi* and *ftdeli* commands in the manual "openFT (Unix and Windows systems) - Command Interface".

Startup and shutdown file (Unix systems)

In openFT on Linux, HP-UX and AIX, there is one global startup and shutdown file that operates on all instances. In addition, every instance present also has its own startup and shutdown file.

During a system startup / shutdown, the global startup and shutdown file is called. This file then calls the startup and shutdown files of all openFT instances.

- Global startup and shutdown file:

It is set up under */etc/init.d* (Linux) or in a corresponding directory on an other Unix platform during the installation of openFT. This startup and shutdown file calls the startup and shutdown files of all instances when the system is started or when it is shut down.

- Startup and shutdown file specific to one instance:

The startup and shutdown file *openFTinst* is created in the */var/openFT/std/etcinit* directory for the *std* instance during the installation of openFT.

If you create another instance with *frcrei*, then a startup and shutdown file *openFTinst* is also set up for this instance. This file is located in the directory *etcinit* of the openFT instance tree.

On Solaris, automatic stop/start is performed via manifests. A manifest is automatically generated for each instance.

4.16 Save and restore 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 partner list, the FTAC environment, and the operating parameter settings in backup files. To do this, you can proceed as follows (the file names used are only examples):

- Back up the partner list using the following command:

```
ftshwptn -px > partner_save (Unix systems)
ftshwptn -pw > partner_save.bat (Windows systems)
```

The file *partner_save* / *partner_save.bat* contains *ftmodptn* commands.

To restore the partner list, simply run the file.

- Back up the FTAC environment (admission sets and profiles) using the following command:

```
ftexpe ftac_save
```

To restore the FTAC environment, import the file using the command `ftimpe ftac_save`.

- Back up the operating parameter settings using the following command:

```
ftshwo -px > option_save (Unix systems)
ftshwo -pw > option_save.bat (Windows systems)
```

The file *option_save* / *option_save.bat* contains an *ftmodo* command.

To restore the operating parameter settings, simply run the file.

- Back up the configuration file of the central administration if necessary:

```
ftexpc remadmin_cfg_save.xml
```

4.17 Characteristics of FTAM on Unix and Windows systems

- Mapping file access rights
 - Outbound requests
 - Inbound requests
- Mapping FTAM attributes to the real file system
 - Inbound mapping of FTAM attributes
 - Inbound mapping the document type
 - Access protection
 - Outbound mapping of the document type

4.17.1 Mapping file access rights

This chapter describes how file protection bits on Unix systems and the access rights on Windows files (FAT and NTFS) are mapped to file management access rights, according to the openFT protocols and as described in the ISO FTAM standard. It provides information on how to modify and display file access rights using the file management functions. A distinction is made here between requests initiated in the local system (outbound) and those initiated in the remote system (inbound).

4.17.1.1 Outbound requests

You can display and modify the file management access rights for files in the remote system.

Display access rights

The access rights for files in the remote system can be displayed using the FT command *ftshw*. The following file management access rights are displayed:

r (read)	read file
p (replace)	overwrite file
x (extend)	extend file
e (erase)	erase data unit (File Access Data Unit FADU), practical for FTAM partners only
a (rdatt)	read file attributes
c (chatt)	change file attributes
d (delete)	delete file

Note on Unix systems

If openFT is installed in the remote Unix system, the file protection bits *r*, *w*, and *x* are mapped to the file access rights as described in the next section for inbound requests.

For FTAM partners, the more restrictive value for access rights, changeable (access control) or unchangeable (permitted actions), is displayed for the respective FTAM partner, since it is relevant for possible file manipulation.

Modify access rights

You can use the FT command *ftmod* to modify file access rights. The access rights of the receive file can also be set or modified for file transfer requests with FTAM. The individual command descriptions indicate which protection bits can be set and how they are to be set in a remote openFT (Unix systems) or openFT (Windows). Access mode options (or combinations of those options) that are not supported are rejected by the file management request, and are ignored by the file transfer request.

4.17.1.2 Inbound requests

Partners in remote systems can display or modify the file management access rights of their own local files.

The access right *i* (insert data unit FADU) is not permitted in Unix and Windows systems.

Display and modify access rights in Unix systems

With a corresponding request from the remote system, openFT (Unix systems) maps the local protection bits *r*, *w*, and *x* to the file management access rights as follows:

Access right displayed	Unix protection bit for the file	Unix protection bit for the parent directories
r (read) read file	r bit	x bit ¹
p (replace) overwrite file	w bit	x bit
x (extend) extend file	w bit	x bit
e (erase) ² erase data unit	w bit	x bit
a (rdatt) read file attribute		x bit
c (chatt) change file attribute	the request must have the same owner authorization as the file	x bit w bit for the next parent directory
d (delete) delete file	w bit	x bit w bit for the next parent directory

¹The r bit of the parent directory is not significant.

²The attribute is practical for FTAM connections only

The access rights of only one user class (owner, group, other) are displayed. The user class is displayed in accordance with the access authorization for the file management request in the Unix system. If a number of user classes have access authorization, the access rights for the highest user class are displayed (e.g. owner access rights before group access rights).

Furthermore, local Unix system rules apply to file access. Thus, for example, the x bit must be set for all parent directories.

Modify access rights in Unix systems

The following table shows the options available in Unix systems for modifying file protection bits:

File management access rights	Unix file protection bits	Function
-------------------------------	---------------------------	----------

rpexecd	rw ¹	read-write
rac	r- ¹	read-only
pxexecd	-w ¹	write-only
ac	-- ¹	none

¹The x bit is not changed by the respective openFT command from the Unix system. From Windows-PCs, even the attributes of remote directories can be changed. In this case, even the x bit is set by rpexecd (=@rw).

The openFT protocols and FTAM only recognize two options for access rights, namely 'set' and 'not set'. This means that when entering access rights, it is necessary to specify whether or not the access right is set. These protocols do not provide the option of leaving access rights unchanged.

To enable file access rights to be modified, the file management access rights *a* and *c* must always be specified; otherwise, the remote request is rejected. If the *w* protection bit is to be set for a file, the file access rights *pxed* must also be set, since all these values are mapped to the *w* file protection bit. All other combinations of file access rights cause the remote request to be rejected.

Only the file owner can modify the access rights of a particular file. Access rights set by the owner can only be modified by the user class 'owner'. However, owner, group, and other user classes can delete access authorizations.

Display access rights in Windows systems

In the case of an appropriate request from a remote system, openFT (Windows) maps the local protection attributes to the file management access rights. A distinction is made between a file on an NTFS file system and a file on a different file system.

The NTFS protection attributes are mapped as follows:

Displayed access rights	File access rights	Rights in the superordinate directory
r (read) read file	read	read
p (replace) overwrite file	read and write	read and write
x (extend) extend file	read and write	read
e (erase) ¹ delete file access data unit	read and write	read
a (read attribute) read file attribute	read	read
c (change attribute) modify file attribute	read and write	read
d (delete) delete file	delete	read and write

¹The attribute is practical for FTAM connections only.

The access rights of the user specified for the file transfer/file management request (directly, indirectly or via a FTAC profile) are valid. The rules for Windows NT apply to the assignment of access rights to a user. If you want to list the attributes of the files in a directory but you (as the calling user) cannot read the protection attributes for several of these files, only the file names are displayed for these files.

Setting the “write protection” attribute (which also exists for FAT files) for an NTFS file has the same effect as not setting the access rights “write” and “delete”.

The FAT protection attributes are mapped as follows:

Displayed access rights	“Write-protection” attribute set
r (read) read file	yes or no
p (replace) overwrite file	no
x (extend) extend file	no
e (erase) delete file access data unit	no
a (read attribute) read file attribute	yes or no
c (change attribute) modify file attribute	no
d (delete) delete file	no

The rights in the superordinate directory are of no significance in FAT systems.

i In Windows systems, no support is provided for really modifying access rights.

4.17.2 Mapping FTAM attributes to the real file system

This section describes the way in which the FTAM implements the virtual filestore, and the mechanisms used for mapping virtual and real filestores in Unix systems oder Windows systems.

Some FTAM attributes are mapped to the attributes available in Unix or Windows systems, and others to the so-called "FTAM catalog (Unix systems) or "FTAM catalog extension (Windows systems).

- The FTAM catalog is used to extend the file attributes available in Unix systems. It is only relevant for access using FTAM. This means that a file can be deleted using the shell command *rm*, even if the *permitted actions* parameter from the FTAM catalog does not permit this for an FTAM partner. This may result in inconsistencies between the FTAM catalog and the real file system. These inconsistencies are detected automatically when openFT (Unix systems) is started and the corresponding entries are deleted from the FTAM catalog.

i The FTAM attributes of a file that are stored in the FTAM catalog are not visible to pointers to the file (such as symbolic links).

- FTAM catalog extension is only possible for NTFS files and enables the file attributes available in Windows to be extended. It is a fixed, but, for the average user, invisible component of a local NTFS file. FTAM catalog extension is not possible for FAT files or files accessed over a net drive or with a UNC name. FTAM-specific attributes are lost if, for instance, an NTFS file is copied in a FAT file system.

FTAM catalog extensions can be generated by:

- File transfer requests with FTAM (on inbound files)
- Inbound file management requests
- Modifying local FTAM attributes

Attributes in an FTAM catalog extension are only of significance if openFT is used for access (generally using FTAM). That is to say, a file can still be deleted with the Windows Explorer or the operating system command *del*, even if the FTAM parameter *permitted actions* does not permit this for an FTAM partner.

Entries in the FTAM catalog / catalog extension 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 or FTAM catalog extension, for example, cannot be transferred as a binary file, nor can it be extended by binary data.

4.17.2.1 Inbound mapping of FTAM attributes

The following tables show how FTAM attributes are mapped to the real file system.

Unix systems

Attribute group	FTAM attributes	Mapping in the Unix system (inbound receive)	Modify FTAM attributes
Kernel group	permitted actions READ-FILE INSERT-DATA-UNIT REPLACE-FILE EXTEND-FILE ERASE-DATA-UNIT READ-ATTRIBUTES CHANGE-ATTRIBUTES DELETE-FILE	FTAM catalog	permitted locally ¹
	universal class number GRAPHIC GENERAL IA5 VISIBLE	FTAM catalog	permitted locally ¹
	string significance VARIABLE FIXED not significant	FTAM catalog	permitted locally ¹
	maximum string length	FTAM catalog	permitted locally ¹
	document type FTAM1 FTAM3	FTAM catalog	permitted locally ¹
Storage group	file availability IMMEDIATE DEFERRED	FTAM catalog	inbound permitted
	future file size	is ignored	not permitted
	storage account	is ignored	not permitted
Security group	ActionList (of 1ACE)		
	READ-FILE	r	inbound permitted
	INSERT-DATA-UNIT	not permitted	not permitted
	REPLACE-FILE	w	inbound permitted
	EXTEND-FILE	w	inbound permitted

ERASE-DATA-UNIT	w	inbound permitted
READ-ATTRIBUTES	x dir	inbound permitted ²
CHANGE-ATTRIBUTES	w dir+owner	inbound permitted ²
DELETE-FILE	w + wdir	inbound permitted
LEGAL-QUALIFICATION	is ignored	not permitted

¹A local modification of the FTAM attribute is possible with the *ftmodf* function.

²The value must always be sent, but may not be changed.

Windows systems

Attribute group	FTAM attributes	Mapping in Windows (inbound)	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 extension	permitted locally ¹
	universal class number GRAPHIC GENERAL IA5 VISIBLE	FTAM catalog extension	permitted locally ¹
	string significance VARIABLE FIXED not significant	FTAM catalog extension	permitted locally ¹
	maximum string length	FTAM catalog extension	permitted locally ¹
	document type FTAM1 FTAM3	FTAM catalog extension	permitted locally ¹
Storage group	file availability IMMEDIATE DEFERRED	FTAM catalog extension	inbound permitted
	future file size	is ignored	not permitted

	storage account	is ignored	not permitted
Security group	ActionList:		
	READ-FILE	Read ²	not permitted
	INSERT-DATA-UNIT	not permitted	not permitted
	REPLACE-FILE	Read + Write ³	not permitted
	EXTEND-FILE	Read + Write ²	not permitted
	ERASE-DATA-UNIT	Read + Write ²	not permitted
	READ-ATTRIBUTES	Read ²	not permitted
	CHANGE-ATTRIBUTES	Read + Write ²	not permitted
	DELETE-FILE	delete ³	not permitted
	LEGAL-QUALIFICATION	is ignored	not permitted

¹A local modification of the FTAM attribute is possible with the *ftmodf* function.

²Read also necessary in parent directory

³Read + write also necessary in parent directory

The following file attributes are derived from the current Unix or Windows file attributes:

- file name
- file size
- identity of creator
- date and time of last read access
- date and time of last attribute modification
- date and time of last modification
- access control

Other attributes are only partially supported by openFT. As the responder, openFT 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 Unix and Windows systems, the FTAM protocol parameter *filestore password* is mapped to the password of the of the login name concerned.

4.17.2.2 Inbound mapping the document type

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

Mapping of the document type for Inbound Receive (FTAM --> Local system)

FTAM (virtual filestore in the remote system)			Local receive file
document type	universal class	string significance	
FTAM-1	25 - GraphicString	variable/fixed	text file
FTAM-1	26 - VisibleString	variable/fixed	text file
FTAM-1	27 - GeneralString	not significant	text file
FTAM-1	22 - IA5String	not significant	text file
FTAM-3	----	not significant	unstructured binary file
FTAM-3	----	variable	record-structured binary file
FTAM-3	----	fix	binary file with fixed record structure

No provision is made for transfer of FTAM-3 files with variable and fixed *string significance* in the functional standard ENV 41204. openFT provides additional support for this function, since the file format corresponds to the user format in Unix and Windows systems.

Mapping of the document type for Inbound Send (FTAM <-- Local system)

FTAM (specifications in request and/or entries in the FTAM catalog / catalog extension in the local system)			Local send file
document type	universal class	string significance	
not specified	not specified	not specified	text file
FTAM-1	not specified	not specified	text file
FTAM-1	25 - GraphicString	variable/fixed	text file
FTAM-1	26 - VisibleString	variable/fixed	text file
FTAM-1	27 - GeneralString	not significant	text file
FTAM-1	22 - IA5String	not significant	text file
FTAM-3	----	not specified	unstructured binary file
FTAM-3	----	not significant	unstructured binary file
FTAM-3	----	variable	record-structured binary file

FTAM-3	----	fix	binary file with fixed record structure
--------	------	-----	---

If there is an entry for the send file in the FTAM catalog / catalog extension, the file format specifications in the request must correspond to this entry. Otherwise, files inconsistencies may occur and file transfer requests involving the particular file may be aborted.

If there are no specifications in the request, the entries in the FTAM catalog / catalog extension apply.

4.17.2.3 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 maps these parameters to the login name and its password in the Unix or Windows system.

For file transfers with FTAM partners it is also possible to use the FTAC functions for extended protection against unauthorized forms of access. If an admission profile in Unix or Windows systems is to be addressed by an FTAM partner, then the 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 FTAC functions 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, or by omitting the file name, etc.).

4.17.2.4 Outbound mapping of the document type

If openFT is the initiator, the FT user can use the file type specification (options *-t*, *-u*, *-b* in *ft* and *ncopy* command) 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 Unix or Windows system.

i In Windows systems, please note that only NTFS files can have entries in the FTAM catalog. No entries can be generated or edited in the FTAM catalog for files on mounted net drives or UNC shares, as mounted systems and UNC shares are handled in the same way as FAT file systems.

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.

Outbound Sending (Local system --> FTAM)

Local system	FTAM attributes)		
File type	document type	universal class	string significance
Text (-t)	FTAM-1	25 - GraphicString	variable ¹⁾
User format (-u)	FTAM-3	----	variable ¹⁾
Binary (-b)	FTAM-3	----	not significant ¹⁾
Binary + record length (-b -r=max record length)	FTAM-3	----	fixed

¹⁾If one of the options *-t*, *-u*, or *-b* are specified and an entry for the send file on the local system exists in the FTAM catalog /catalog extension, this entry must correspond to the entries in the above table.

If the FT user does not specify a file type in the request, the entries in the FTAM catalog are used. If there is no entry in the FTAM catalog, FTAM1, GraphicString, and variable are used.

No provision is made for transfer of FTAM-3 files with variable *string significance* in the functional standard ENV 41204. openFT systems provides additional support for this function.

Outbound Receive (Local system <-- FTAM)

For outbound receive, the type of the receive file depends on whether and which file type, if any, was specified in the FT request. The following cases must be differentiated here.

1. No file is specified in the request

FTAM (virtual filestore in the remote system)			Receive file on the local system
document type	universal class	string significance	
FTAM-1	25 - GraphicString	variable/fixed	text file
FTAM-1	26 - VisibleString	variable/fixed	text file

FTAM-1	27 - GeneralString	not significant	text file
FTAM-1	22 - IA5String	not significant	text file
FTAM-3	----	not significant	unstructured binary file
FTAM-3	----	variable	record-structured binary file
FTAM-3	----	fix	binary file with fixed record structure

2. *-t* option resp. *Text Format* specified for file type in request

FTAM (virtual filestore in the remote system)			Receive file on the local system
document type	universal class	string significance	
FTAM-1	25 - GraphicString	variable/fixed	text file
FTAM-1	26 - VisibleString	variable/fixed	text file
FTAM-1	27 - GeneralString	not significant	text file
FTAM-1	22 - IA5String	not significant	text file

3. *-u* option resp. *User format* specified for file type in the request

FTAM (virtual filestore in the remote system)			Receive file on the local system
document type	universal class	string significance	
FTAM-3	----	variable	record-structured binary file
FTAM-3	----	fix	binary file with fixed record structure

4. *-b* option resp. *Binary* specified for file type in the request

FTAM (virtual filestore in the remote system)			Receive file on the local system
document type	universal class	string significance	
FTAM-3	----	not significant	unstructured binary file

5. *-b* and *-r* (max. record length) options resp. *Binary + Maximum Record Length* specified for file type in the request

FTAM (virtual filestore in the remote system)			Receive file on the local system
document type	universal class	string significance	
FTAM-3	----	fix	binary file with fixed record structure

5 Central administration

Central administration in openFT covers the functions

- Remote administration
- ADM traps

5.1 Remote administration

- Configuring the remote administration server
 - Defining the ADM administrator
 - Declaring an openFT instance as a remote administration server
 - Setting up admission profiles for accessing the remote administration server
 - Entering the openFT instances to be administered in the partner list
 - Creating a configuration file using the Configuration Editor
 - Creating a configuration file using a text or XML editor
 - Importing the configuration
 - Exporting and modifying a configuration
- Configuring an openFT instance to be administered
 - Configuring an admission profile for an openFT instance as of V11.0
 - Configuring an admission profile for an openFT instance < V11.0
- Issuing remote administration requests
 - Remote administration using the command interface
 - Remote administration using the openFT Explorer
- Logging remote administration

5.1.1 Configuring the remote administration server

The remote administration server must be configured in a number of steps. Some of these steps can only be performed by the ADM administrator, who must have been defined beforehand.

Overview of the configuration steps

openFT as of V11.0 must be installed on your system if it is to be configured as a remote administration server. The description in the present subsection applies to openFT V12.1.

The following table indicates the steps required to create a configuration and who performs these steps.

Step	Who
1. Defining the ADM administrator	FTAC administrator
2. Declaring an openFT instance as a remote administration server	FT administrator
3. Setting up admission profiles for accessing the remote administration server	ADM administrator
4. Entering the openFT instances to be administered in the partnerlist	FT administrator
5. Creating a configuration file using a text or XML editor	ADM administrator
6. Importing the configuration	ADM administrator

The remote administration server is thus ready for operation. The ADM administrator can export and modify the current configuration at any time. See [section “Exporting and modifying a configuration”](#).

It now remains to configure openFT instances on the partner systems for remote administration. See [section “Configuring an openFT instance to be administered”](#).

5.1.1.1 Defining the ADM administrator

The ADM administrator is the only person permitted to administer the remote administration server. Because no ADM administrator is defined by default after openFT has been installed, we urgently recommend that you define one first. This property is bound to the admission set and must therefore be assigned by the FTAC administrator.

In your role as FTAC administrator, call the following command:

```
ftmoda userid -admpriv=y
```

This makes the user ID *userid* the ADM administrator. Once the ADM administrator has been defined, only the ADM administrator is permitted to transfer the permission to another user ID or return the admission using `ftmoda -admpriv=n`. It is not sufficient for you to be an FT administrator or an FTAC administrator.

If you do not specify a user ID (`ftmoda -admpriv=y`) you are both the FTAC administrator and the ADM administrator.

The ADM administrator is indicated in the ATTR column in the output from the *ftshwa* command. The value ADMPR appears in the associated admission set.

In place of the commands you can also use the openFT Explorer functions, for instance via the object directory *Admission Sets* in the object tree on the left-hand side or using the menu: *File - New - Admission Set*.

5.1.1.2 Declaring an openFT instance as a remote administration server

To allow an openFT instance to act as a remote administration server, this must be specified explicitly in the operating parameters of the instance.

To do this, the FT administrator enters the following command:

```
ftmodo -admcs=y
```

Alternatively, you can set this operating parameter using menu system of the openFT Explorer: *Administration - Operating Parameters, Protocols* tab, *Remote Administration Server* option.

i

- As soon as an openFT instance is declared as a remote administration server, the operating parameter *Administration Connections* is implicitly changed and set to 64! If a high load is to be expected, the FT administrator can increase this value, in particular if the openFT instance is also used as an ADM trap server. See [section “Configuring the ADM trap server”](#).
- For reasons of performance, it is recommended that a separate computer that only handles remote administration tasks and that possibly also acts as the ADM trap server is used as the remote administration server.

5.1.1.3 Setting up admission profiles for accessing the remote administration server

To ensure that the remote administrators obtain access to the remote administration server, the ADM administrator must set up special admission profiles with the property "Access to Remote Administration Server" (ACCESS-TO-ADMINISTRATION). The owner of these admission profiles is always the ADM administrator, and never the remote administrator for whom access using such a profile is set up.

It is urgently recommended that you set up a separate admission profile for each remote administrator in order to make it clear which remote administrator has made changes to which openFT instance.

As ADM administrator, enter the command `ftcrep` with the option `-ff=c`:

```
ftcrep profile-name transfer-admission -ff=c
```

`profile-name`

Identifies this profile name. You must enter this name in the configuration file when you define the remote administrator. See [section "Creating a configuration file using a text or XML editor"](#).

`transfer-admission`

Identifies the FTAC transfer admission. The remote administrator must specify this with a remote administration request. See [section "Remote administration using the command interface"](#).

In addition, for reasons of security, you can use `-pn=part1,part2,...,partn` to specify the partner(s) from which a remote administrator is permitted to access the remote administration server.

You can also set up the profile using the openFT Explorer by making the following settings in the *Options* tab of the *Admission Profile* dialog box:

- Activate the option *Access to Remote Administration Server*.
- Deactivate all file transfer functions under *Permissible FT Functions*.

5.1.1.4 Entering the openFT instances to be administered in the partner list

On the remote administration server, the FT administrator should enter the openFT instances that are to be administered in the partner list. This makes it possible to reference the instances using the names in the partner list, which has the following benefits:

- If the address changes, it is only necessary to change the entry in the partner list. This avoids the necessity of modifying and re-importing the configuration file.
- It is possible to explicitly use partner checking and authentication, thus eliminating security risks on the path between the remote administration server and the administered openFT instance.

The FT administrator enters the partners in the partner list. To do so, use the command *ftaddptn*. Alternatively, you can use the openFT Explorer to navigate to the object directory *Partner List* in the object tree, for instance, and choose *New Partner List Entry...* from the context menu.

Address format of the partners

Partners using openFT as of V11.0 and openFT < V11.0 have different address formats.

- Partners using openFT as of V11.0 must be entered as ADM partners. An ADM partner has the following address format:

```
ftadm: //host[:port number]
```

port number only needs to be specified if the default ADM port (11000) is not used on the computer *host* of the instance to be administered.

- Partners using openFT < V11.0 must be entered as openFT partners, because the *ftexec* command is used internally for remote administration:

```
host[:port number]
```

port number only needs to be specified if the default openFT port (1100) is not used on the computer *host* of the instance to be administered.

i The ADM administrator must additionally specify the attribute *Mode="Legacy"* in the configuration file for such partners. See the [section "Creating a configuration file using a text or XML editor"](#).

5.1.1.5 Creating a configuration file using the Configuration Editor

This section is intended for **ADM administrators**.

With the Configuration Editor, openFT provides a graphical interface which you can use to create or edit configuration files. The configuration file is an input file in XML format in which the ADM administrator defines the following:

- the remote administrators
- the openFT instances and groups of instances to be administered by these remote administrators
- the remote administration rights that the remote administrators have on each of the openFT instances (access list)

You must then import this file, see [section “Importing the configuration”](#).

The representation of the configuration corresponds to the display you will subsequently see under *Remote Administration* in the openFT Explorer, see the example under [section “Creating a configuration file using the Configuration Editor”](#).

Creating a new configuration file

The most important steps are described below. See the online help for detailed information on the dialog boxes and the individual parameters.

1. Start the openFT Explorer.
2. Start the Configuration Editor by opening the *Extras* menu and choosing the *Start Configuration Editor* command.

You then see the Configuration Editor start window.

3. Open the *File* menu and choose the *New Configuration* command.

The *Configuration* node is displayed in the navigation area. Here you define the individual objects in the configuration using the commands in the context menu.

- Administrators

For the first administrator, choose the *New Administrator* command in the context menu of the *Configuration* node. Define the properties in the *Administrator* dialog box.

Repeat this step for each administrator that you want to define.

- Groups

Select the *New Group* command in the context menu of the *Configuration* node and define the corresponding properties in the *Group* dialog box.

Repeat this step for all the other groups that you want to define.

For each group, you can also create subgroups by selecting the *New Group* command from the context menu of a group.

- Instances

Select the *New Instance* command from the context menu. You can select this command in the *Configuration* node (creates an instance at the topmost level) or in a node corresponding to a group (creates an instance within a group). You define the attributes of the instance in the *Instance* dialog box.

Repeat this step for all the other instances that you want to define.

- Access lists

You can create access lists for the entire configuration (global access list), for groups or for individual instances:

Select the *Create Access List* context menu command. You can choose this command in the *Configuration* node (global access list, in the node corresponding to a group (applies to all the instances in a group, including the instances in the subgroups) or of an instance.

When you do this, only the *Access List* item is initially created. Now choose the *New Access Entry* context menu command under *Access List* and define the corresponding access rights in the *Access Entry* dialog box.

i You can use the *Properties* command in an access list's context menu to open the *Access List* dialog box. Here you can specify whether access permissions are to be inherited from parent access lists. This dialog box also displays any access permissions that may have been inherited.

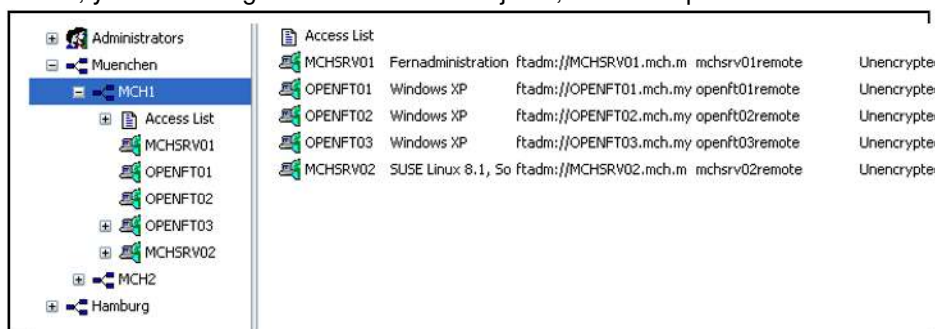
4. Finally, save the entire configuration using the

Save as command in the *File* menu. At save time, openFT checks the validity of the configuration file. If any errors are detected, you will see a corresponding message together with a query asking you whether you want to save the file anyway.

Modifying the configuration file

You can use the Configuration Editor to modify an existing configuration irrespective of how this was created.

1. Start the Configuration Editor in the same way as if you were creating a new configuration file.
2. Open the *File* menu and choose *Open*.
3. In the *Open Configuration File* dialog box that is now opened, select the file containing the configuration that you want to modify.
4. The configuration is displayed in the navigation area in the form of a tree structure. By expanding the individual nodes, you can navigate to each of the objects, see example below:



5. You can use the context menu commands to add new objects (in the same way as when creating a new configuration file). In addition, you can

- Modify an object's attributes:

Choose the *Properties* command in the object's context menu. The properties can be modified in the dialog box that is now opened for the object.

-
- Move objects:
Choose the *Copy* or *Cut* command from an object's context menu, navigate to the required position and choose the *Paste* command from the context menu. Alternatively, you can also use the mouse to move objects in the navigation area (corresponds to *Cut + Paste*).
 - Delete objects:
Choose the *Delete* command from the object's context menu (alternatively: *Del* key). You must always explicitly confirm the delete operation
6. Finally, save the modified configuration by opening the *File* menu and choosing *Save* (overwrites the old configuration file) or *Save as*.

5.1.1.6 Creating a configuration file using a text or XML editor

This section is intended for **ADM administrators**.

The configuration file is an input file in XML format in which the ADM administrator defines the configuration. In principle, you can create the file on any system using a text editor. It is, however, advantageous if you work on the (future) remote administration server and use an XML editor, for instance, the free XML editor "XML Notepad 2007" from Microsoft. If you do this, you can use the supplied template, complete with schema so that your entries are immediately checked. See [section "Creating a configuration file using a text or XML editor"](#).

Describing the configuration data in XML format provides a simple way to represent a complex configuration clearly by forming groups.

In the configuration file, you define:

- the configuration, see [section "Creating a configuration file using a text or XML editor"](#),
- the remote administrators, see [section "Creating a configuration file using a text or XML editor"](#),
- the openFT instances and groups of instances to be administered by these remote administrators, see [section "Creating a configuration file using a text or XML editor"](#),
- the remote administration permissions that the remote administrators have on each of the openFT instances (access list), see [section "Creating a configuration file using a text or XML editor"](#),
- the encoding mode for ftadm commands, see [section "Creating a configuration file using a text or XML editor"](#).

The ADM administrator must then import the configuration file into the remote administration server using the *ftimp* command. See [section "Importing the configuration"](#). The *ftexp* command allows you to create an XML file from the internal configuration data again at any time, in order to modify the configuration, for instance.

The structure of the XML file is described in the following sections. An exhaustive example is given in the [section "Example of an XML configuration file"](#).

Using the XML template and XML schema

The directory *samples/ftadm* (Unix systems) or *samples\ftadm* (Windows systems) under the openFT installation directory contains the file *config.xml*, which contains a simple sample configuration that can be used as a template and adapted appropriately.

The schema on which the XML file is based is defined in the file *config.xsd*, which is located in the *include* directory of openFT after installation. If you are using an XML editor, you can use the file *config.xml* as the basis for your work. The installation path of the schema file *config.xsd* is entered in this file. This means that the XML editor uses this schema in order to immediately verify your entries. If *config.xsd* has been copied elsewhere or renamed, you must adjust the installation path of *config.xsd* in *config.xml*.

Defining the configuration

The configuration file contains precisely one configuration for a remote administration server. It is structured hierarchically, i.e. child elements are nested inside a parent element.

A configuration starts with the XML tag `<Configuration>` and comprises the following attributes:

- Mandatory attribute *Version*. The value of the attribute *Version* is a string that specifies the version of the configuration data. The maximum length of the string is 4 bytes. In openFT V12.1, "1210" must be specified for the version.

- Optional attribute *Description*. The value of the attribute *Description* is a string that describes the configuration data in more detail. The maximum length of the string is 100 bytes.

Example:

```
<Configuration
  Version="1210"
  Description="Configuration for central server MCHSRV01">
  <...
  .../>
</Configuration>
```

Elements of a configuration

A configuration contains the following elements:

- At least one *administrator ID* element with the tag <AdministratorID> for defining a remote administrator. You can define up to 100 remote administrators. For a detailed description, refer to the [section “Creating a configuration file using a text or XML editor”](#).
- Optional *access list* element with the tag <AccessList>. You use an access list to define the administration permissions on the openFT instances for the individual remote administrators. For a detailed description of the access list, refer to the [section “Creating a configuration file using a text or XML editor”](#).
- Optional *group* elements with the tag <Group>. Groups can be nested, thus allowing the geographical or organizational structure of a company to be represented, for instance. The maximum nesting depth is limited. See the note in [section “Creating a configuration file using a text or XML editor”](#). For a detailed description of a group, refer to [section “Creating a configuration file using a text or XML editor”](#).
- At least one *instance* element with the tag <Instance> for the openFT instances. You can define up to 5000 instances. For a detailed description of an instance, refer to [section “Creating a configuration file using a text or XML editor”](#).

i A pathname is formed from the name of the instance and the name of the group (where appropriate with subgroups) according to the following pattern:

```
group/subgroup1/subgroup2/.../instance
```

The remote administrator must enter precisely this pathname in a remote administration request to the instance. See also [section “Remote administration using the command interface”](#).

This pathname can be a maximum of 200 characters long. The maximum number of subgroups therefore depends on the lengths of the individual names.

Defining remote administrators

In the configuration file, you specify which remote administrators are permitted to perform remote administration. To do this, proceed as follows:

- Define one or more remote administrators
- Assign each remote administrator a profile name and/or a user ID on the remote administration server.

A remote administrator is defined using the XML tag <AdministratorID>. You can enter a maximum of 100 remote administrators in the XML file. The <AdministratorID> tags must be defined immediately following the <Configuration> tag, because the subsequent definitions for the groups and instances reference them.

<AdministratorID> has the following attributes:

- Mandatory attribute *Name*. The value of the attribute *Name* is a string that specifies the name of the remote administrator. The maximum length of the string is 32 bytes. The name must be unique, i.e. the configuration file must not contain any other <AdministratorID> tags with the same name. The name is used both internally in the configuration data and externally in log records in order to uniquely identify the initiator of a remote administration request.
- Optional attribute *Description*. The value of the attribute *Description* is a string that describes the remote administrator in more detail. The maximum length of the string is 100 bytes.
- Optional attributes *UserID* and *Profile*. These attributes identify the remote administrator depending on the type of access to the remote administration server. You must therefore specify a least one of the two attributes *UserID* or *Profile*. It is also possible to enter both attributes.

The following applies to *UserID* and *Profile*:

- The value of the *UserID* attribute is a string with the name of a valid login ID on the remote administration server. The maximum length of the string depends on the platform and can be up to 36 bytes.

The user that logs in on the remote administration server locally under this ID is therefore a remote administrator and possesses the administration permissions granted to this *AdministratorID*. A particular login ID must therefore only be specified for one *AdministratorID*, otherwise the correlation between the user ID <-> remote administrator is no longer unique.

- The value of the *Profile* attribute is a string with the name of a valid FTAC profile. The maximum length of the string is 8 bytes. The ADM administrator of the remote administration server must be the owner of the profile. Each FTAC profile name may only be used with exactly one *AdministratorID*.

This profile is used if the remote administrator issues a remote administration request on a remote computer and sends it to the remote administration server using the FTADM protocol. In this event, the remote administrator must specify the associated transfer admission in the request.

The profile must include the function ACCESS-TO-ADMINISTRATION (corresponds to *ftcrep -ff=c*) See [section "Setting up admission profiles for accessing the remote administration server"](#).

Example:

```
<Configuration
  Version="1210">
  <AdministratorID
    Name="John"
    Description="Domain Controller Administrator"
    UserID="rz\John"
    Profile="Profile01"/>
  <AdministratorID
    Name="Fred"
    Profile="Profile02"/>
  <...
    .../>
</Configuration>
```

Defining groups and openFT instances to be administered

The configuration file contains all the openFT instances that can be administered via this remote administration server using the remote administration facility.

Defining groups

By defining groups and subgroups with freely selectable names, it is possible to organize the openFT instances that are to be administered in a way that meets your precise requirements. When groups are formed, the path of an instance is made up of the *Name* attributes of the parent groups and the instance in question, e.g. *Muenchen/MCH1/OPENFT01*. The complete pathname must not exceed a total length of 200 bytes. The maximum nesting depth therefore depends on the lengths of the individual names.

A group starts with the XML tag <Group>. There is no limit to the maximum number of groups in the XML file. The groups must be defined **after** the remote administrators in the XML file, because the subsequent definitions for the groups and instances reference the remote administrators.

A group is made up of the following attributes:

- Mandatory attribute *Name*. The value of the attribute *Name* is a string that specifies the name of the group. The maximum length of the string is 24 bytes and it may not contain a slash (/). The name could, for instance, be the name of a town, a branch office or a department, or it could simply be the description of the functions of a group of openFT instances.
- Optional attribute *Description*. The value of the attribute *Description* is a string that describes the group in more detail. The maximum length of the string is 100 bytes.

The following elements can be assigned to a group:

- Optional *access list* element with the tag <AccessList>. You use the access list to define for the individual remote administrators the remote administration permissions on the openFT instances that belong to this group and to any subsequent child groups. For a detailed description of the access list, refer to [section “Creating a configuration file using a text or XML editor”](#).
- Optional *group* elements with the tag <Group>. You can specify any number of groups. By specifying further nested groups, it is possible to represent the relationships between the groups hierarchically. In this event, the total path length must not exceed 200 bytes. See the note in [section “Creating a configuration file using a text or XML editor”](#).
- Optional *instance* elements with the tag <Instance> for the openFT instances that belong to this group. You can define up to 5000 instances in a single configuration.

i Specification of the *group* and *instance* elements within a group is optional, but a group must contain at least one further group or one instance.

Example:

```

<Configuration
  ...>
  <AdministratorID
    .../>
  <Group
    Name="Muenchen"
    Description="Computer Center Muenchen">
    <Group
      Name="MCH1"
      Description="Computer Center Muenchen Schwabing">
      <AccessList>
        <AccessEntry
          .../>
      </AccessList>
    </Group>
  </Group>

```

```

  <Instance
    Name="MCHSRV01"
    ... />
  <Instance
    Name="OPENFT01"
    ... />
</Group>
<Group
  Name="MCH2"
  Description="Computer Center Muenchen Freimann">
  ...
</Group>
...
</Group>
...
</Configuration>

```

Defining instances

An openFT instance starts with the XML tag <Instance>. You can define a maximum of 5000 instances in the XML file.

An instance can be assigned to a group or defined independently of a group. You must observe the following assignment hierarchy:

- With group(s):

Configuration

Remote administrator(s)

Optional access list

Group(s):

Optional access list

Instance

Optional instance-specific access list

- Without group:

Configuration

Remote administrator(s)

Optional access list

Instance

Optional instance-specific access list

You will find detailed information on the access list in [section “Creating a configuration file using a text or XML editor”](#).

An instance is made up of the following attributes:

- Mandatory attribute *Name*. The value of the attribute *Name* is a string that specifies the name of the openFT instance. The maximum length of the string is 24 bytes and it may not contain a slash (/). The name of the instance can be freely selected.
- Optional attribute *Description*. The value of the attribute *Description* is a string that describes the instance in more detail. The maximum length of the string is 100 bytes.
- Mandatory attribute *Address*. The value of the attribute *Address* is a string with a maximum length of 200 bytes that specifies the address of the openFT instance to be administered. You can specify the name from the partner list or enter the address directly.

The address format of the administered openFT instance depends on its version:

- openFT as of V11.0:
The address must have the protocol prefix *ftadm://*, i.e. it must be entered with this prefix in the partner list or the prefix must be specified here. If this is not done, the openFT instance will be administered as an openFT instance < V11.0 using *ftexec*.
- openFT < V11.0:
The address must have the standard format, i.e. it must be entered without a prefix in the partner list or the prefix must not be specified here. You must also set the *Mode* attribute to the value “*Legacy*”. See below.
- Mandatory attribute *Admission*. The value of the attribute *Admission* is a string containing the FTAC transfer admission. The maximum length of the string is 36 bytes (67 bytes if specified in hexadecimal format). An admission profile with this transfer admission must be defined in the openFT instance to be administered. Depending on the version of the instance to be administered, this profile must permit the following function(s). See the [section “Configuring an openFT instance to be administered”](#):
 - openFT as of V11.0:
REMOTE-ADMINISTRATION (corresponds to *ftcrep ... -ff=a*)
 - openFT < V11.0:
TRANSFER-FILE + FILE-PROCESSING (corresponds to *ftcrep ... -ff=tp*)

i If there are separate FT and FTAC administrators in the openFT instance that is to be administered then enter one of the two transfer admissions (for the FT administrator or FTAC administrator) for the *Admission* attribute. If necessary, you may have to create a second instance with the other transfer admission.

- Optional Attribute *Mode*. The string “*Legacy*” can be specified for the *Mode* attribute. This means that the openFT instance is an instance < V11.0 that can only be administered using *ftexec*. In this case, no protocol prefix *ftadm://* is allowed to be specified in the partner address.

- Optional Attribute *DataEncryption*. The string "Yes" can be specified for the *DataEncryption* attribute. This means that the user data exchanged between the remote administration server and the openFT instance to be administered is transferred in encrypted form. If the *DataEncryption* attribute is missing, the user data is not encrypted when it is transferred.

DataEncryption="Yes" can only be specified if openFT-CR is installed both on the remote administration server and on the instance that is to be administered.

An instance can contain the following element:

- Optional access list with the tag `<AccessList>`. The access list allows you to define nonstandard permissions for individual remote administrators that only apply to this instance. You can extend or restrict the inherited permissions or deactivate inheritance and specify other permissions. For a detailed description of the access list, refer to the [section "Creating a configuration file using a text or XML editor"](#).

Example:

```
...
  <Group
    Name="MCH1"
    Description="Computer Center Muenchen Schwabing">
      <AccessList>
        <AccessEntry
          .../>
      </AccessList>
      <Instance
        Name="MCHSRV01"
        Description="Remote administration server"
        Address="ftadm: //MCHSRV01.mch.mycompany.net"
        Admission="mchsrv01remote"/>
      <Instance
        Name="OPENFT01"
        Description="Windows 7"
        Address="ftadm: //OPENFT01.mch.mycompany.net:11009"
        Admission="openft01remote">
        <AccessList>
          <AccessEntry
            .../>
        </AccessList>
      </Instance>
    </Group>
  ...
```

Defining an access list

In the access list, you specify which remote administrators have access to the given openFT instance to be administered and what remote administration permissions are granted to each of the remote administrators.

The following rules apply:

-
- An access list can be defined at the following locations:
 - before all groups and/or instances. The list then applies to all subsequent groups and/or instances provided that separate access lists have not been defined for these.
 - as an element of a group. The list then applies to all openFT instances that belong to this group and is inherited by all child groups.
 - as an element of an openFT instance that is to be administered. The list then only applies to this instance.
 - Every openFT instance that is to be administered requires an access list that is either defined explicitly with the instance or that is inherited from parent elements (associated group, parent group or an access list defined before all groups/instances).

An openFT instance without an access list (access lists) that has been either explicitly set or implicitly inherited cannot be administered.

- You can explicitly control the scope of inheritance in an access list of a child group or for an openFT instance:
 - You can deactivate inheritance using the optional attribute *InheritFromParent*. In this event, you must define a separate access list for this instance in which you specify the administration permissions for the remote administrators.
 - You can expand or restrict inherited permissions for particular remote administrators (*AllowFunction* and *DenyFunction* attributes under `<AccessEntry>`). Entries which deny a function to a specific remote administrator take priority over entries that permit a function for a specific remote administrator. Additional entries in access lists for groups are also inherited by child groups.

Elements of an access list

An access list starts with the XML tag `<AccessList>`. There is no limit to the maximum number of access lists in the configuration file. The access list can be defined at different places in the file. See [section “Creating a configuration file using a text or XML editor”](#).

An access list has the following attribute:

- Optional attribute *InheritFromParent*.
The value of the attribute *InheritFromParent* can accept the string *No*. If *No* is specified, inheritance of access lists from parent groups is deactivated. Because access lists are inherited from parent groups by default, it is only necessary to specify the attribute *InheritFromParent* if inheritance is to be explicitly deactivated.

An access list can contain the following element:

- one or more *access entries* with the XML tag `<AccessEntry>`.
Any number of access entries is permitted. However, an access list may contain a maximum of one access entry for each remote administrator. An access entry allows you to explicitly define the access permissions for a remote administrator. This means that you can specify which remote administration functions are granted or denied to this remote administrator.

Note that parent access permissions are inherited unless you have deactivated this by specifying *InheritFromParent="No"*.

Defining an access entry

An access entry is an element of an access list and starts with the XML tag `<AccessEntry>`. There is no limit to the maximum number of access entries in the configuration file. An access entry is made up of the following attributes:

- Mandatory attribute *AdministratorID*. The value of the attribute *AdministratorID* is a string that specifies the name of the remote administrator. This remote administrator must be defined at the start of the configuration file using the tag `<AdministratorID>`. See [section “Creating a configuration file using a text or XML editor”](#). A remote administrator may only be specified in one access entry in an access list.
- *AllowFunction* and *DenyFunction* attributes. These attributes specify which remote administration functions are granted (*AllowFunction*) and denied (*DenyFunction*). The *AllowFunction* and *DenyFunction* attributes are in principle optional, but you must specify at least one of the two attributes in every access entry.

If both attributes are specified, note that entries for the attribute *DenyFunction*, which deny a function to the remote administrator, take priority over entries for the attribute *AllowFunction*, which grant this function to the remote administrator.

The following points apply:

- The value of the attribute *AllowFunction* specifies what remote administration functions are permitted to the remote administrator to carry out. The string can have the following values (remote administration permissions):

```
"FTOP"  
"FT"  
"FTAC"  
"FT FTAC"  
"FTAC FT"  
"FTAC FTOP"  
"FTOP FTAC"
```

- Specifying *"FTOP"* (FT operator) only permits read FT access.
- Specifying *"FT"* permits FT access for reading and modification.
- Specifying *"FTAC"* permits FTAC access for reading and modification.

Combinations mean that the remote administrator has been granted both permissions.

- The value of the attribute *DenyFunction* determines which remote administration functions have been denied to the remote administrator. The string can have the following values:

```
"FT"  
"FTMOD"  
"FTAC"  
"FT FTAC"  
"FTAC FT"  
"FTAC FTMOD"  
"FTMOD FTAC"
```

- Specifying *"FTMOD"* denies FT access for modification.
- Specifying *"FT"* denies FT access for reading and modification.
- Specifying *"FTAC"* denies FTAC access for reading and modification.

Combinations mean that both functions are denied.

This means, for example, that *"FTAC FTMOD"* means that neither FTAC access nor FT access for modification is permitted. In other words, read FT access only is permitted, which corresponds to specifying *"FTOP"* under *AllowFunction*.

Example:

```
<Group
  Name="HH1"
  Description="QA Computer Center">
  <AccessList>
    <AccessEntry
      AdministratorID="Jack"
      AllowFunction="FT FTAC"/>
    <AccessEntry
      AdministratorID="Mike"
      AllowFunction="FT FTAC"/>
  </AccessList>
  <Instance
    Name="HHWSRV02"
    Description="HP-11"
    Address="ftadm: //HHWSRV02.hhw.mycompany.net"
    Admission="hhwsrv02remote"/>
  <Instance
    Name="HHWSRV11"
    Description="Solaris 10"
    Address="HHWSRV11.hhw.mycompany.net"
    Admission="hhwsrv11remote"
    Mode="Legacy">
    <AccessList>
      <AccessEntry
        AdministratorID="Mike"
        DenyFunction="FTAC"/>
    </AccessList>
  </Instance>
</Group>
```

Defining encoding mode

The attribute *CmdMode* defines a recommendation in the remote administration server for every administered instance as to whether character or transparent mode is to be used. The openFT desktop of the administrator can take its bearings from these specifications after a *ftshwc* call to the central server. *CmdMode* does not force you to use a specific mode, but helps avoid errors in the process and achieve optimal behavior.

CmdMode can be set to one of the two following values:

- *Transparent*:
The central remote administration recommends the use of transparent mode (*ftadm -fnc=t*)
- *Char*:
The central remote administration recommends the use of character mode (*ftadm -fnc=c*)

CmdMode can be specified in the XML configuration of the remote administration server in three different places:

- as a parameter in <Configuration>: This specification is the default setting for all instances and groups, unless there are individual settings there.
- as a parameter in the <Group>: This specification overwrites default settings from the configuration or from superior groups and applies for all groups and instances within this group, unless these have individual settings.

-
- as a parameter in the <Instance>: This specification has priority over all default settings. If this is an instance that works in legacy mode, the transparent mode is automatically assumed - even if nothing is explicitly specified for this instance. If a *CmdMode* specification is not found in an instance, which does not work in legacy mode, neither in the instance itself nor in one of the groups, to which the instance belongs, nor in the configuration, then there is no mode recommendation.

If the *CmdMode* attribute is set, the version in the configuration must be 1210 or higher. openFT V12.1 can import XML configuration files of the previous versions (with the version specification 1100, for example); but the version has to be set to 1210 for export.

5.1.1.7 Importing the configuration

The configuration defined in the configuration file still has to be converted to the internal, optimized format, which in turn activates it.

To do this, the ADM administrator enters the command *ftimpc* at the remote administration server:

```
ftimpc xml-file
```

xml-file identifies the configuration file that you have created previously. See [section “Creating a configuration file using a text or XML editor”](#).

Alternatively, you can perform this action in the openFT Explorer: *Administration* menu, *Remote Administration - Import Configuration...* command.

The file can be imported during live operation.

After the configuration file has been imported, the remote administration server is ready for operation. It is able to accept remote administration requests and forward them to the openFT instances to be administered.

5.1.1.8 Exporting and modifying a configuration

openFT provides the ADM administrator with an export function that allows the configuration data to be backed up, checked or modified.

It is not possible to change the configuration data directly on the remote administration server.

i Note that the purpose of the *ftshwc* command is not to output the entire configuration for the ADM administrator. Its purpose is rather to show a remote administrator the openFT instances which that administrator is able to administer, including the remote administration permissions on the instances that have been granted to the administrator.

Exporting the configuration

If the ADM administrator wishes to export the configuration, he/she must enter the following command on the remote administration server:

```
ftexpc xml-file
```

Alternatively, in the openFT Explorer:

Administration menu, *Remote Administration - Export Configuration...* command.

The configuration data is stored in XML format in the file *xml-file*. The notation is the same as is used when creating the configuration file. See [section “Creating a configuration file using a text or XML editor”](#).

The file can be exported during live operation.

Changing the configuration

The following steps are necessary if the ADM administrator wishes to change a configuration, for instance in order to add instances or change addresses:

1. Export the configuration into a file as described above, e.g. using *ftexpc xml-file*.
2. Make the changes in the file. For details, see [section “Creating a configuration file using the Configuration Editor”](#) or [section “Creating a configuration file using a text or XML editor”](#).
3. Import the changed file, e.g. using *ftimpc xml-file*. See also [section “Importing the configuration”](#).

The configuration can be imported during live operation. If, however, the changes to the configuration are particularly extensive, a message is issued prompting you to stop the asynchronous openFT server before performing the import. You can use the commands *ftstop* and *ftstart* or the corresponding commands in the *Administration* menu of the openFT Explorer to stop and subsequently start the server.

The changes take effect immediately. However, running ADM requests with the old configuration are not canceled. The new configuration is displayed in the openFT Explorer if you choose the *Update* command from the context menu of the relevant remote administration server.

5.1.2 Configuring an openFT instance to be administered

The remote administration server uses FTAC transfer admissions to access the openFT instances. These must be entered in the configuration file when defining the openFT instance. See [section “Creating a configuration file using a text or XML editor”](#).

This means that the appropriate admission profiles must be defined in the openFT instances from which administration is being carried out. The properties of this profile depend on the version of the openFT instance to be administered.

5.1.2.1 Configuring an admission profile for an openFT instance as of V11.0

To allow remote administration, an admission profile with the function "Remote Administration" (REMOTE-ADMINISTRATION) must be set up on the instance to be administered. The following cases must be distinguished:

- An admission profile with the permission FT (FT access for reading and modification) or FTOP (FT access for reading) must belong to the FT administrator.
- An admission profile with the permission FTAC (FTAC access for reading and modification) must belong to the FTAC administrator.
- An admission profile with the permission FT+FTAC (FT and FTAC access for reading and modification) can only be set up if the FT administrator is also an FTAC administrator. If this is not the case, two profiles must be created (for FT and for FTAC). The instance must then also be configured twice in the configuration file of the remote administration server, once for FT remote administration and once for FTAC remote administration.

Example

The FT administrator enters the following command for an admission profile, for instance:

- Unix or Windows system:

```
ftcrep profile-name transfer-admission -ff=a
```

Possible alternative using the openFT Explorer: Open the *Admission Profile* dialog box, for instance using *File - New - Admission Profile*, and then in the *Options* tab, activate the option *Remote Administration via Remote Administration Server*.

- BS2000 system:

```
CREATE-FT-PROFILE NAME=profile-name -  
                ,TRANSFER-ADMISSION=transfer admission -  
                ,FT-FUNCTION=*REMOTE-ADMINISTRATION
```

- z/OS:

```
FTCREPRF NAME=profile-name -  
        ,TRANSFER-ADMISSION=transfer admission -  
        ,FT-FUNCTION=*REMOTE-ADMINISTRATION
```

If you also wish to ensure that this profile can only be used by a particular remote administration server, specify this using *-pn=server* (Unix and Windows system) or *PARTNER=server* (BS2000 system and z/OS).

5.1.2.2 Configuring an admission profile for an openFT instance < V11.0

To allow remote administration, an admission profile must be set up on the instance to be administered that permits the FT functions "Transfer Files" (TRANSFER-FILE) and "Pre/Postprocessing" (FILE-PROCESSING). The same comments apply as for an openFT instance as of V11.0 (see [section "Configuring an admission profile for an openFT instance as of V11.0"](#)).

Example

The FT administrator enters the following command for an admission profile, for instance:

- Unix or Windows system:

```
ftcrep profile-name transfer-admission -ff=tp
```

Possible alternative using the openFT Explorer: Open the *Admission Profile* dialog box, for instance using *File - New - Admission Profile*, and then in the *Options* tab, activate the options *Transfer Files and/or Delete Files* and *File Processing*.

- BS2000 system:

```
CREATE-FT-PROFILE NAME=profile-name -  
    ,TRANSFER-ADMISSION=transfer admission -  
    ,FT-FUNCTION= (*TRANSFER-FILE, *FILE-PROCESSING)
```

- z/OS:

```
FTCREPRF NAME=profile-name -  
    ,TRANSFER-ADMISSION=transfer admission -  
    ,FT-FUNCTION= (*TRANSFER-FILE, *FILE-PROCESSING)
```

5.1.3 Issuing remote administration requests

This section is intended for all **remote administrators** for whom specific permissions for remote administration have been specified in the configuration of the remote administration server.

Remote administrators can perform remote administration using commands (see below) or using the openFT Explorer (see [section “Remote administration using the openFT Explorer”](#)).

You can issue the requests on the remote administration server itself or on a remote computer:

- If you issue requests on the remote administration server, you must log in under the user ID that the ADM administrator has entered in the configuration data to authenticate yourself as a remote administrator.
If you log in on the remote administration server under a user ID that is not entered in the configuration data, you can only address the remote administration server using the FTADM protocol. This is the same as if you issue the request on a remote computer. See the next section.
- If you issue requests on a remote computer, you require the following data that the ADM administrator must provide you with:
 - address of the remote administration server
 - FTAC transfer admission for accessing the remote administration server

The address of the remote administration server must always be specified with the protocol prefix *ftadm://*, e.g. *ftadm://server01*. It is therefore always best to let the FT administrator enter the remote administration server in the partner list.

You are, however, always able to determine the names of the openFT instances that you are permitted to administer yourself. See the [section “Remote administration using the command interface”](#).

5.1.3.1 Remote administration using the command interface

If you use the command interface for remote administration, you must first determine the names of the openFT instances that you are permitted to administer.

Determining the names of the openFT instances

You obtain the names of the openFT instances using the command *ftshwc*. You can enter the command directly on the remote administration server. On a remote computer, you must "package" it using the command *ftadm*:

- Entering *ftshwc* on the remote administration server:

```
ftshwc -rt=i
```

- Entering *ftshwc* on the a remote computer:

```
ftadm -cs=server "ftshwc -rt=i" transfer-admission
```

Explanation

server

Name of the remote administration server from the partner list or address of the remote administration server using the format *ftadm://host...* .

transfer-admission

FTAC transfer admission for accessing the remote administration server. The associated profile must have the property ACCESS-TO-ADMINISTRATION (see [section "Setting up admission profiles for accessing the remote administration server"](#)) and the profile name must be assigned to a remote administrator in the configuration file (see [section "Creating a configuration file using a text or XML editor"](#)).

Sample output

```
TYPE      = *INSTANCE      ACCESS = FT+FTOP+FTAC / *CHAR
NAME      = Muenchen/Jonny
DESC      = Computer Test-en-lp
TYPE      = *INSTANCE      ACCESS = FTOP
NAME      = Muenchen/Hello
DESC      = Computer Hello
```

NAME specifies the name of the instance that you must specify exactly as given here in the remote administration request. Your remote administration permissions for this instance and - if configured - the encoding mode are listed under ACCESS, see also the description of *ftshwc*.

Issuing a remote administration request

You issue a remote administration request using the *ftadm* command.

The syntax used for the remote administration request depends on whether you enter the *ftadm* command directly on the remote administration server or on a different, remote computer.

-
- Entering the *ftadm* command on the remote administration server:

Log in on the remote administration server under the user ID that the ADM administrator has configured as remote administrator in the configuration file. See the *UserID* attribute in the [section “Creating a configuration file using a text or XML editor”](#).

Enter the *ftadm* command in the following form:

```
ftadm -ri=instance "command"
```

- Entering the *ftadm* command on a remote computer:

Log in on the remote computer using any user ID and enter the *ftadm* command in the following format:

```
ftadm -cs=server  
      -ri=instance "command" transfer-admission
```

Explanation

server

On the remote computer only: Name of the remote administration server from the partner list or address of the remote administration server using the format

ftadm: //host... .

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 with the *ftshwc* command. See [section “Remote administration using the command interface”](#).

command

Specifies the administration command to be executed on the openFT instance. You should always enclose *command* in quotes. If *command* contains spaces or special characters, the quotes are mandatory.

transfer-admission

On the remote computer only: FTAC transfer admission for accessing the remote administration server. The associated profile must have the property ACCESS-TO-ADMINISTRATION (see [section “Setting up admission profiles for accessing the remote administration server”](#)) and the profile name must be assigned to a remote administrator in the configuration file (see [section “Creating a configuration file using a text or XML editor”](#)).

5.1.3.2 Remote administration using the openFT Explorer

The object tree of the openFT Explorer contains the item *Remote Administration* with the following icon:



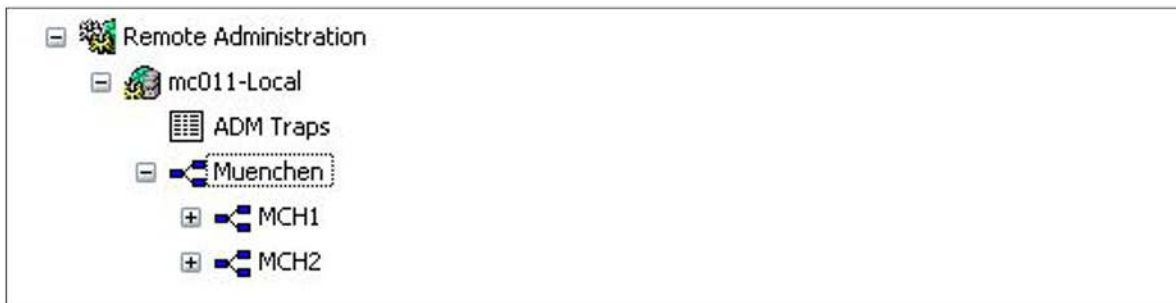
You can log in to the remote administration server locally or perform remote administration from a remote computer.

Logging into the remote administration server locally

If you log in to the remote administration server locally and your user ID is configured as a remote administrator there, the object tree displays an additional icon for the local remote administration server.

The local remote administration server has the name *server-name* -Local, where *server-name* is the host name of the remote administration server.

If you click on this node, all openFT instances that you are permitted to administer are displayed.



Local administration server

In this example, the group *Muenchen* is shown with the two subgroups MCH1 and MCH2 that you are permitted to administer.

Performing remote administration from a remote computer

If the remote administration server is on a different computer, you must first set it up in the openFT Explorer. In addition, the FT administrator should also enter it in the partner list.

The following steps are required:

- Entering the remote administration server in the partner list

The FT administrator enters the remote administration server in the partner list using the following format:

```
ftadm://host[:port number]
```

port number only needs to be specified if the default ADM port (11000) is not used on the remote administration server *host*. The same applies if you, as the remote administrator, specify the address directly in a remote administration request.

- Entering a remote administration server in the openFT Explorer
 1. Choose *New Remote Administration Server...* from the context menu of the *Remote Administration* object directory in the object tree.
 2. Enter the following details in the *Remote Administration Server* dialog box:
 - The partner (where possible the name from the partner list).
 - The FTAC transfer admission for accessing the remote administration server. The associated profile on the remote administration server must have the property ACCESS-TO-ADMINISTRATION (see [section “Setting up admission profiles for accessing the remote administration server”](#)) and the profile name must be assigned to a remote administrator in the configuration file (see [section “Creating a configuration file using a text or XML editor”](#)).

If you also activate the *Save Transfer Admission* option, this has the advantage that you do not have to specify the transfer admission in future every time you call the openFT Explorer.

When you click *OK*, a new icon appears in the object tree with this remote administration server.

Clicking on the name of the remote administration server opens the associated object directory. In the example below, an additional server *remadmin* is set up alongside the local remote administration server *mc011-Local* (see [section “Remote administration using the openFT Explorer”](#)).



Local and remote administration servers in the openFT Explorer

Issuing remote administration requests

All instances that can be administered are listed under the relevant groups (in the example, these are *Muenchen* and *Hamburg*). The context menu of an instance allows you to access the operating parameters and diagnostics information of the instance and view the properties.

If you expand the subtree of an instance, the icons for all the administration objects of the instance are displayed:



Administration objects of an instance in the openFT Explorer

You can administer these objects of the instance (*OPENFT01* in the example) in the same way as you would normally do locally with openFT. For further details, refer to the online Help system. In addition, you can access the trace files for the instance via the *Traces* object directory.

5.1.4 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 with the *ftshwl* command,
- and you can delete ADM log records with the *ftdell* command provided that you have the appropriate permission.

Alternatively, you can also view and delete ADM log records using the openFT Explorer (*Logging* object directory in the object tree).

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

Do this using the *ftmodo -la* command or the openFT Explorer (*Administration - Operating Parameters* menu, *Logging* tab).

5.2 ADM traps

- Configuring the ADM trap server
- Configuring ADM traps in the openFT instance
- Viewing ADM traps

5.2.1 Configuring the ADM trap server

To allow an openFT instance to act as an ADM trap server, you must carry out the following actions in your role as FT administrator:

- The "Remote Administration Server" function must be activated on the ADM trap server. To do this, enter the command `ftmodo -admcs=y`.
Alternatively: In the openFT Explorer, choose *Administration - Operating Parameters* to open the *Addresses* tab, and activate the option *Remote Administration Server*.

It is not necessary for an ADM trap server to be simultaneously used as a remote administration server, but this does have the advantage that every remote administrator can view "their" ADM traps using the remote administration facility. See [section "Viewing ADM traps"](#).

- In the ADM trap server, set up an admission profile that can be used for the administration function "Receive ADM traps". To do this, use the `ftcrep` command with the `-ff=l` option.
Alternatively: In the openFT Explorer open the *Options* tab in the *Admission Profile* dialog box and activate the *Receive ADM traps* option.

The transfer admission for this profile must be entered in the operating parameters of the openFT instances that are to send the traps to the ADM trap server. See "[Configuring ADM traps in the openFT instance](#)".

The ADM traps are stored in the file `sysatpf`, which is located in the `log` directory of the relevant openFT instance. In the case of the standard instance, the pathname is:

- `/var/openFT/std/log/sysatpf` (Unix systems)
- `%ProgramData%\Fujitsu Technology Solutions\openFT\var\std\log\sysatpf` (Windows systems)

The file `sysatpf` is written cyclically. This means that the oldest ADM trap entry is overwritten when a given maximum size is exceeded.

ADM traps cannot be explicitly deleted.

5.2.2 Configuring ADM traps in the openFT instance

To enable an openFT instance to send ADM traps to the ADM trap server, the FT administrator of the openFT instance must make certain settings in the operating parameters, see below. In addition, the asynchronous openFT server must be started.

The procedure for Unix and Windows systems is described below. You will find the descriptions for BS2000 and z/OS systems in the manual "openFT (BS2000) - Installation and Operation" and in the manual "openFT (z/OS) - Installation and Operation".

Carry out the following actions in your role as FT administrator:

- Specify the following items in the `-atpsv` option of the `ftmodo` command:
 - the name of the ADM trap server:
The ADM trap server must be an ADM partner, i.e. it must either be defined in the partner list using the address format `ftadm: //host...` or the address must be specified directly using the format `ftadm: //host... .`
 - the transfer admission for the admission profile defined in the ADM trap server for this purpose. See [section "Configuring the ADM trap server"](#).
- In the `-atp` option of the `ftmodo` command, you specify the events on which ADM traps are to be sent to the ADM trap server:
 - state change of the asynchronous openFT server
 - Change of partner status
 - Unavailability of partners
 - Change of request management status
 - Successfully completed requests
 - Failed requests

i For reasons of performance, you should restrict the scope of the ADM traps to the necessary minimum, for instance to failed requests or the unavailability of partners. If, for example, ADM traps for all successfully completed requests are sent to the ADM trap server by several instances, this can place a heavy load on the local openFT system, the ADM trap server and the network.

Alternatively, you can also perform these actions using the openFT Explorer:

1. Choose *Administration, Operating Parameters...* to open the *Traps* tab.
2. In the *ADM Trap Server* group, enter the name of the ADM trap server and the transfer admission.
3. In the *ADM* column of the *Type* group, select the events on which ADM traps are to be sent.

5.2.3 Viewing ADM traps

The FT administrator of the ADM trap server is permitted to view the ADM traps. If the ADM trap server is also used as the remote administration server, both the ADM administrator and the remote administrators can view traps.

The following points apply:

- If you log in to the ADM trap server as an FT administrator or ADM administrator, you can view all ADM traps. There are two ways of doing this:
 - Using the *ftshwatp* command. In this case you can select traps according to different criteria (source, period, number, etc.).
 - Using the openFT Explorer: Under *Administration* in the object tree, click *ADM Traps* (see figure) or choose *Show ADM Traps* from the context menu of the alarm icon (if present) in the status bar:



Viewing ADM traps in the openFT Explorer as the FT administrator

You can set the selection criteria using the context menu. The ADM traps are shown in the form of a list in the openFT Explorer.

For further details, refer to the online Help 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 [section "Remote administration using the command interface"](#). The following options are available:

- If you log in directly on the remote administration server, enter the command `ftshwatp`.

Alternatively: In the openFT Explorer, under *Remote Administration* in the object tree, click *ADM Traps* for the local server.

- If you log in on a remote computer, enter the following command:

```
ftadm -cs=server "ftshwatp options" transfer-admission
```

Explanation

options

ftshwatp command options which you use to define the selection criteria for ADM traps and the output format. If you do not specify any options then the most recent ADM trap is output in short format.

server

Name of the remote administration server from the partner list or address of the remote administration server using the format *ftadm: //host...*

transfer-admission

FTAC transfer admission for accessing the remote administration server. The associated profile must have the property ACCESS-TO-ADMINISTRATION (see [section "Setting up admission profiles for accessing the remote administration server"](#)) and the profile name must be assigned to a remote administrator in the configuration file (see [section "Creating a configuration file using a text or XML editor"](#)).

Alternatively, using the openFT Explorer: In the object tree under *Remote Administration*, open the object directory of the remote administration server and click *ADM Traps*. See the figure below:



Viewing ADM traps in the openFT Explorer using the remote administration facility

You can set the selection criteria using the context menu. The ADM traps are shown in the form of a list in the openFT Explorer.

For further details, refer to the online Help system.

5.3 Example of an XML configuration file

The configuration for the company *mycompany* is made up of four computer centers, two in Munich (MCH1, MCH2) and two in Hamburg (HH1, HH2). A separate subgroup is created for each computer center. The remote administration computer MCHSRV01 is located in MCH1.

Four remote administrators are configured: *John, Fred, Jack* and *Mike*. The following table shows the groups, subgroups and openFT instances and specifies which remote administrator has which permissions.

Group	Subgroup	Instance	Permissions of the remote administrator			
			John	Fred	Jack	Mike
Muenchen	MCH1	MCHSRV01	FT	FT, FTAC		
		OPENFT01	FT	FT, FTAC		
		OPENFT02	FT	FT, FTAC		
		OPENFT03	FTOP	FT, FTAC		
	MCHSRV02			FT, FTAC		
	MCH2	MCHSRV03	FT, FTAC			
Hamburg	HH1	HHWSRV01			FT, FTAC	FT, FTAC
		HHWSRV02			FT, FTAC	FT, FTAC
		HHWSRV11			FT, FTAC	FT
	HH2	HHWSRV99			FT, FTAC	FTOP

XML configuration file

The configuration shown in the table is defined using the following configuration file. Items indicated by numbers on the right margin are explained after the file.

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration
  Version="1210"
  Description="Configuration for central server MCHSRV01">
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  ; Only on Unix systems
  xsi:noNamespaceSchemaLocation="/opt/openFT/include/config.xsd">
  ; Only on Windows systems
  xsi:noNamespaceSchemaLocation="<openFT-installationdirectory>/include/config.xsd">

  <AdministratorID
    Name="John"
```

```

    Description="Domain Controller Administrator"
    UserID="rz\John" 1.
    Profile="Profile01"/> 2.
<AdministratorID
    Name="Fred"
    Description="Production computer administrator"
    UserID="rz\Fred" 1.
    Profile="Profile02"/> 2.
<AdministratorID
    Name="Jack"
    Description="Administrator of the HR department computer in HH"
    Profile="Profile03"/> 2.
<AdministratorID
    Name="Mike"
    Description="Administrator of the QA computer in HH"
    Profile="Profile04"/> 2.
<Group
    Name="Muenchen"
    Description="Computer Center Muenchen">
  <Group
    Name="MCH1"
    Description="Computer Center Muenchen Schwabing">
      <AccessList> 3.
        <AccessEntry
          AdministratorID="John"
          AllowFunction="FT"/>
        <AccessEntry
          AdministratorID="Fred"
          AllowFunction="FT FTAC"/>
      </AccessList>
    <Instance
      Name="MCHSRV01" 4.
      Description="Remote administration server"
      Address="ftadm: //MCHSRV01.mch.mycompany.net"
      Admission="mchsrv01remote"/>
    <Instance
      Name="OPENFT01" 4.
      Description="Windows 10"
      Address="ftadm: //OPENFT01.mch.mycompany.net"
      Admission="openft01remote"/>
  </Group>
</Group>

```

```

<Instance
  Name="OPENFT02"
  Description="Windows 10"
  Address="ftadm: //OPENFT02.mch.mycompany.net "
  Admission="openft02remote"/>
<Instance
  Name="OPENFT03"
  Description="Windows 10"
  Address="ftadm: //OPENFT03.mch.mycompany.net "
  Admission="openft03remote">
  <AccessList>
    <AccessEntry
      AdministratorID="John"
      DenyFunction="FTMOD"/>
    </AccessEntry>
  </AccessList>
</Instance>
<Instance
  Name="MCHSRV02"
  Description="SUSE SLES 11, source management"
  Address="ftadm: //MCHSRV02.mch.mycompany.net "
  Admission="mchsrv02remote">
  <AccessList>
    InheritFromParent="No">
    <AccessEntry
      AdministratorID="Jack"
      AllowFunction="FT FTAC"/>
    </AccessEntry>
  </AccessList>
</Instance>
</Group>
<Group
  Name="MCH2"
  Description="Computer Center Muenchen Freimann">
  <AccessList>
    <AccessEntry
      AdministratorID="John"
      AllowFunction="FT FTAC"/>
    </AccessEntry>
  </AccessList>
  <Instance
    Name="MCHSRV03"
    Description="Windows Server 2016 domain controller"
    Address="ftadm: //MCHSRV03.mch.mycompany.net "
    Admission="mchsrv03remote">
  </Instance>
</Group>

```

```

</Group>
<Group
  Name="Hamburg"
  Description="Computer Center North in Hamburg Wandsbek">
  <Group
    Name="HH1"
    Description="QA Computer Center">
    <AccessList> 3.
      <AccessEntry
        AdministratorID="Jack"
        AllowFunction="FT FTAC"/>
      <AccessEntry
        AdministratorID="Mike"
        AllowFunction="FT FTAC"/>
    </AccessList>
    <Instance
      Name="HHWSRV01" 4.
      Description="Solaris 10"
      Address="ftadm: //HHWSRV01.hhw.mycompany.net"
      CmdMode="Char"
      Admission="hhwsrv01remote"/>
    <Instance
      Name="HHWSRV02" 4.
      Description="HP-11"
      Address="ftadm: //HHWSRV02.hhw.mycompany.net"
      Admission="hhwsrv02remote"/>
    <Instance
      Name="HHWSRV11" 4.
      Description="Solaris 10"
      Address="HHWSRV11.hhw.mycompany.net"
      Admission="hhwsrv11remote"
      Mode="Legacy"> 6.
      <AccessList> 5.
        <AccessEntry
          AdministratorID="Mike"
          DenyFunction="FTAC"/>
        </AccessList>
      </Instance>
    </Group>
  </Group>

```

```

<Group
  Name="HH2"
  Description="HR department">
  <AccessList>
    <AccessEntry
      AdministratorID="Jack"
      AllowFunction="FT FTAC"/>
    <AccessEntry
      AdministratorID="Mike"
      AllowFunction="FTOP"/>
  </AccessList>
  <Instance
    Name="HHWSRV99"
    Description="Mainframe system (BS2000)"
    Address="ftadm: //HHWSRV99.hhw.mycompany.net"
    Admission="hhwsrv99remote"/>
  </Group>
</Group>
</Configuration>

```

Explanation

1. User ID that has the specified administrator permissions on the remote administration server. This allows remote administration to be performed directly on the remote administration server. If no user ID is specified here, remote administration is only possible using the FTAC transfer admission (see 2).
2. Name of the admission profile for accessing the remote administration server. The profile must include the function ACCESS-TO-ADMINISTRATION (corresponds to *ftcrep -ff=c*). If remote administration is performed from a remote computer, the remote administrator must specify the associated FTAC transfer admission.
3. Defines the admissions for the entire group. An `<AccessEntry>` tag is specified for each authorized remote administrator. This permission can be expanded or restricted in an instance (see 5).
4. Defines an instance. The complete address (as in the example) or the name from the partner list can be specified in the *Address* attribute. Partners with openFT as of V11.0 must be defined with *ftadm: //...*
Admission specifies the transfer admission for the instance to be administered. The associated admission profile must be set up there and must permit the REMOTE-ADMINISTRATION function (Corresponds to *ftcrep -ff=a*). In addition, the encoding mode can be defined using *CmdMode*.
5. The `<AccessList>` tag for an instance defines permissions that only apply for this instance:
 - The *InheritFromParent="No"* attribute cancels a parent (inherited) permission.
 - The *DenyFunction* attribute under `<AccessEntry>` restricts inherited permissions. For instance, the *FT* permission is reduced to *FTOP* with *DenyFunction="FTMOD"*.
 - *AllowFunction* defines or extends permissions.
6. The *Mode="Legacy"* attribute specifies that an openFT version < V11.0 is running on the instance. The instance is addressed as an openFT partner, i.e. the address is specified without a prefix. The *ftexec* command is then used internally for a remote administration request.

6 Troubleshooting and Diagnosis

This chapter contains the following information:

- What if ...
- Creating diagnostic records
- Trace function
- Additional diagnostic information
- Code tables

6.1 What if ...

... the message "Local file is inconsistent" is output.

This may be because

- a binary file was inadvertently transferred as a text file (Use the *-b* option!)
- a text file contains records that are too long (Use the *-r* option!)

... the message "Remote system not available" is output?

This may be because

- the partner address specified in the partner list, TNS or hosts entry is not correct. For BS2000 interconnections, you should check whether a BCMAP entry for \$FJAM was made with the port number 1100 on the BS2000 partner (this is automatically created as of openFT V9.0 for BS2000 system).
- the asynchronous openFT server has not started on the partner system.
- a firewall in the partner system is blocking connections.

i You can perform a test to see whether *ftping <partner address>* results in a response from the remote openFT system.

Please note that *ftping* is only intended for internal use and does not represent a guaranteed interface.

... the local system cannot be reached from the partner systems?

The following potential error sources should be examined:

- Is the asynchronous openFT server started?
- Does the local address match the default settings (*ftmodo -openft=@s*) or has it been changed?
- If you use TNS on a Windows system:
 - Was an RFC1006 entry with TSEL \$FJAM made for the local address?
 - Was port number 1100 assigned to the local application \$FJAM? Port number 102 should basically never be used, since this could result in collisions with other application packets.
- Was port number 1100 addressed in the partner system? In BS2000 systems, openFT automatically generates a BCMAP. For this to be successful, no old BCMAP entries may be present.
- Is the openFT application released on the firewall?

... the message "Local system unknown in remote system" is output?

This means that your partner system does not accept your local system as a partner. In this case, you should check the following on the partner system:

- Are dynamic partners connected and is there no or no suitable entry in the partner list for your local system?
Possible solutions:
 - Enter your local system in the partner list on the partner system or
 - Check the partner list entry in the remote system to determine, for example, whether the sent instance identification matches the entered instance identification, or
 - Permit dynamic partners.

-
- Does partner address checking fail for your local system?

Check the settings for the operating parameters *Identification* and *Processor* on the local system.

... the message "Remote system xy unknown" is output?

This may be because

- you must change the partner list entry, the TNS entry or the entry in the hosts file for the partner system,
- a TNS entry is being used even though the use of TNS has been deactivated,
- dynamic partners have been deactivated and the partner is not entered in the partner list.

... the BS2000 system cannot be accessed

If your local system in BS2000 is unknown, enter the command ADD-FT-PARTNER in BS2000.

If you receive the message "Remote system not available", check whether one of the following reasons is the cause:

- Resource bottleneck in the remote system
- Remote FT system is not started
- BCIN is missing
- no network connection (for a TCP/IP connection, check the connection with the command *ping*, for example)
- Name server entry is missing or is incorrect

... the name of the partner is missing in the log records

Enter the partner in the partner list, in the DNS, in the hosts file (*/etc/hosts* on Unix systems) or in the TNS.

... the logging function cannot be called, i.e. the logging file is no longer readable or is inconsistent

Possible reasons are:

1. System crash while log records are being written.
2. File system full while writing the logging file.
3. In Unix systems: *kill* on the openFT process while log records are being written.

The only remedy here is to terminate openFT (*ftstop*) and delete the log file.

You can determine the full path name of the log file in question using the command *ftshwl -llf -plf=0*, providing that the log file has not been changed since the problem occurred.

On Windows systems you can, for instance, use the Windows Explorer in order to delete the log file.

This means that you lose all log records in the affected file.

The explicit creation of an empty log file is not reasonable because an inconsistent log file remains due to missing header information.

To prevent space problems, you should

- regularly change the log file (*ftmodo -lf=c*),
- back up old log files on another computer or storage medium
- and then delete the old offline log files on the openFT computer.

Alternatively: Activate the automatic deletion of log records (*ftmodo*, options *-ld*, *-lda*, *-ldd* and *-ldt*).

... access to the admission set and admission profile file causes errors or if this file is defective

The possible reasons are:

1. Manual access to *sysfsa.dat* and/or *sysfsa.idx*. These files are located in the respective openFT instance directory under *config*.

Unix system:

The path name of these files is as follows with the standard instance:

/var/openFT/std/config/sysfsa.dat

and

/var/openFT/std/config/sysfsa.idx

2. System crash with *sysfsa.** open
3. In Unix systems, *kill* of openFT process with *sysfsa.** open
4. In Unix systems, file system full on ISAM access

In Unix systems in cases 2 and 3 and 4, ISAM generally leaves an unusable index file.

Possible solutions:

- Attempt to export/import:
Use *ftexpe* to export the data to a backup file.
Then shut down the openFT server with *ftstop*, delete *sysfts.dat* and *sysfsa.idx* and restart openFT with *ftstart*.
Import the data from the backup file using *ftimpe*.

- Try to repair the ISAM index file with *dcheck*.

Example valid for the standard instance on Unix systems:

```
/opt/openFT/bin/ftbin/dcheck -b /var/openFT/std/config/sysfsa
```

Example valid for the standard instance on Windows 10 systems:

```
dcheck -b "C:\ProgramData\Fujitsu Technology  
Solutions\openFT\var\std\config\sysfsa"
```

It may be necessary to delete the index file explicitly:

- If the data file *sysfsa.dat* is empty then no data is lost. As a result, both ISAM files can be deleted with openFT stopped and can then be initialized before *ftstart* by using the *ftshwa* command.
- If the data file already contains modifications to the admission sets and/or profiles then you should enter the following commands:

Unix system:

```
cd /var/openFT/std/config  
ftstop  
mv sysfsa.dat sav.sysfsa.dat && rm sysfsa.idx  
ftshwa >/dev/null  
rm sysfsa.dat && mv sav.sysfsa.dat sysfsa.dat  
/opt/openFT/bin/ftbin/dcheck -b sysfsa  
ftstart
```

Windows system:

```
cd C:\ProgramData\Fujitsu Technology Solutions\var\std\config  
ftstop  
move sysfsa.dat sav.sysfsa.dat  
del sysfsa.idx  
ftshwa  
del sysfsa.dat  
move sav.sysfsa.dat sysfsa.dat  
dcheck -b sysfsa  
ftstart
```

Explanation:

If *sysfsa.idx* is defective, it must be recreated. To do this, you must first back up the *sysfsa.dat* file that you want to create. You then use *ftshwa* to create a new *sysfsa.dat* file which you immediately delete and replace with the backed up *sysfsa.dat* file. The resulting file pair can now be re-used.

- If this attempt also fails, you must delete the admission set and admission profile and make new entries to ensure a consistent state.

... You are not given a free transport connection for an ncopy request

- On Windows systems this may occur with a connection to a non-TCP/IP network (e.g. X.25). Check the configuration settings for the corresponding transport system.
- Check the partner address in the partner entry or in the partner list.
- If you are working with TNS: check your TNS entries and check whether TNS use and operation with CMX are activated (in the case of *ftshwo*, the value YES must be displayed for USE TNS and USE CMX; otherwise activate TNS use and operation with CMX with *ftmodo -tns=y -cmx-y*).

-
- Check the address settings in the operating parameters.

... the openFT message “Remote transfer admission invalid” appears

For reasons of data security, this message does not differentiate between the various possible reasons for the rejection on the initiator side. This information is only available via the openFT logging of the responder system.

... requests still remain in the “WAIT” state?

- Check whether the asynchronous openFT server is started in the local system
- Check whether the openFT or asynchronous openFT server is started in the remote system

Using *ftshwr -l*, you can obtain further information on the cause.

.. in Unix systems, deleting a request in the openFT Explorer takes an unusually long time (about 1 minute)

This may mean

- that a request was issued to send a mail when the request to be deleted is finished
- and that the mail function of the Unix system takes about 1 minute to send a mail due to a configuration problem.

Solution:

Do not ask for a mail to be sent when the request is finished, i.e. specify the *-m=n* option for the *ft* command (or omit *-m* because *-m* is default as of V10.0). Requests that are started in the openFT Explorer never require a mail to be sent when finished.

... in Linux systems, the left mouse button does not function as desired in the openFT Explorer

This may be due to the fact that the function of the NumLock key was set differently on generation with Xfree and KDE (in larger SuSE Linux systems).

This causes problems if the NumLock key functions as an Alt Lock key: a click then becomes an Alt-click and a double-click becomes an Alt-double-click.

The administrator can overcome this problem by toggling the NumLock key. It may also be possible to set the Numlock functionality in the BIOS. The *xmodmap* command can be used to check and modify the keyboard allocation.

... on a Windows Terminal Server or Windows Server, the message "Can't create termination event (error x). Command aborted." is output when a user executes an openFT command.

This means that the openFT command cannot be executed due to missing user privileges. The problem might occur on Windows Terminal Server or Windows Server if the privilege to "Create global objects" is not granted to the "Users" group. The System Administrator must therefore grant the privilege "Create global objects" to the "Users" group to solve the problem.

... in Windows systems, the openFT service only starts when the system is rebooted, but cannot be started manually although the user has the necessary administrator rights?

In this case you will receive an error message from Windows with number 0xC0000022 regarding a failed initialization. This happens when a path with a network drive or UNC name or a path containing spaces has been entered in the PATH system environment variable before the openFT installation path. If the service starts automatically when the system is booted, then these entries are not active yet and the service will start normally. They will then be activated later on, but SYSTEM will not be able to access them because it does not have the proper rights.

Solution:

Clean up the path.

... in Windows systems, initialization errors occur in user32.dll or kernel32.dll when follow-up processing is started?

Cause:

The system environment variable PATH contains path inaccessible UNC paths/network drives.

Solution:

Clean up the path and use only local accessible paths.

Performance note

If you use the TNS during operation with CMX (*ftmodo -tns=y*), you should set the RFC1006 protocol for TNS entries, since the RFC1006 protocol is far more efficient than communicating via LANINET. In BS2000 systems, you should work without BMAP entries. If you nevertheless need BMAP entries then the following applies: If the PTSEL-I entry exists, RFC1006 is used.

In the case of operation with CMX, the RFC1006 protocol is always used.

6.2 Creating diagnostic records

If, in spite of precautions, an error occurs which neither the FTAC administrator nor the system administrator can rectify, please contact your local Fujitsu Technology Solutions contact partner. In order to simplify error diagnosis, you should provide the following documents:

- an exact description of the error situation and information as to whether the error is reproducible;
- the version number of the file transfer product in the own system;
- the version number of the file transfer product in the remote system, and the operating system of the remote system;
- diagnostic information, see [section “Additional diagnostic information”](#)
- if available, the FTAC, FT and ADM log records (which are output with the FT command *ftshwl ...*);
- if available, the openFT trace file;
- for errors related to a specific admission profile a printout of the profile (*ftshwp'BLANK'profilename'BLANK'-l*) and a printout of the admission sets (*ftshwa'BLANK'@a*).
- the version and the variant of the operating system
- the version of the communication system (CMX, etc.)
- if necessary, the process tables (*ps* command on Unix systems)

On Unix systems you can also call the procedure `/opt/openFT/bin/ftbin/ftdiaginfo` to initiate the collection of various diagnostic data. This procedure generates the file `ftdiaginfo.tgz` (compressed tar file) and saves it in the current directory. Send this file together with a description of the error to the responsible contact person.

On Windows systems, the program `diaginfo.exe` is available for this purpose.

6.3 Trace function

This section describes how you can create and evaluate trace files.

6.3.1 Trace files

You can switch trace mode on or off for the purposes of error diagnosis.

6.3.1.1 Activating/deactivating trace functions

You can control the trace function as follows:

- You use the command `ftmodo -tr=n/f` to activate and deactivate the trace function itself.
- When the trace function is active, the command `ftmodo -trp -trr` allows you to make selections based on protocol type and request type.
- You use the command `ftmodo -tro=b` to generate a minimal trace.
- You use the command `ftmodo -troll` to control the scope of the trace for the lower protocol layers.

This means that it is also possible to store CMX trace files in the instance's directory when working with CMX. These can be selected and displayed in the same way as openFT trace files using the openFT Explorer, for example.

You can also make these settings in the openFT Explorer (*Administration - Operating Parameters - Trace*).

You can also create a partner-specific trace, see [section "Activating/deactivating trace functions"](#).

When trace mode is switched on, diagnostic data is written to trace files which are located in subdirectory *traces* of the respective openFT instance.

In the case of the standard instance the path name is as follows:

- On Unix systems `/var/openFT/std/traces`
- On Windows systems `C:\ProgramData\Fujitsu Technology Solutions\openFT\var\std\traces`

When you have finished diagnosis, you should deactivate the trace mode for reasons of performance. The trace files can become infinitely large, since they are not cyclically overwritten. However, you can also close trace files with the `ftmodo -tr=c` command and open new trace files. This function is also available in the openFT Explorer (*Change File* button on the *Trace* tab).

Activating partner specific trace

If you only wish to record traces for a specific partner, proceed as follows:

1. Activate the trace function for the required partner, for instance using `ftmodptn partner1 -tr=n`.
2. Deactivate the trace for the partner types, for instance using `ftmodo -trp=`.
3. Deactivate the general trace function, for instance using `ftmodo -tr=n`.

Activating/deactivating interface trace (Windows systems)

You can additionally activate the interface trace using the openFT Explorer. To do this, proceed as follows:

1. Activate the *Interface Trace* option under *Administration - Operating Parameters - Trace*.
2. Stop the openFT service using the control panel and then restart it.

Deactivation is performed in the same way:

1. Deactivate the *Interface Trace* option under *Administration - Operating Parameters - Trace*.
2. Stop the openFT service using the control panel and then restart it.

i Note that the interface trace is extremely extensive and can slow down operation of openFT. For this reason, you should only activate the interface trace if this is required for diagnostic purposes.

6.3.1.2 Viewing trace files

You can either view trace files directly in the openFT Explorer or open them in an editor after preparing them with the *fttrace* command. The trace files are located in the directory *traces* of the respective openFT instance.

Files which have the suffix *.fttf* are prepared directly and are displayed in the openFT editor when double clicking on such a file in the openFT Explorer.

Files with the suffix *.fttf* are protocol trace files. Their names begin with *Y* or *S*. In Windows systems, files with the suffix *.PPE* are interface trace files.

The names of the trace files have the following format:

- *Yoddhmm.Sssccc.Pppppp.fttf*
Protocol trace files for synchronous outbound requests.
- *Soddhmm.Sssccc.l000.fttf*
Protocol trace files for the control process.
- *Soddhmm.Sssccc.liii.fttf*
Protocol trace files for the server processes that handle asynchronous outbound requests and inbound requests.
- *process-pid-thid-time.PPE* (Windows systems)
Interface trace files. Here, *process* is the name of the process which the command has executed, *pid* the process ID as a hexadecimal number, *thid* the thread ID as a hexadecimal number and *time* the time in milliseconds since the system start.

Explanation for protocol trace files

oddhmm.Sssccc

specifies the creation time of the protocol trace file. Here, *o* 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).

ppppp

specifies the Process ID of the protocol trace file if Type=Y.

iii

is the index of the server process (type=S), starting with 001.

Trace files in case of errors

- If a trace file cannot be written without errors due to a memory bottleneck, a message to this effect is output.
- If a record of a server process 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.l001.fttf (first trace file)
S8101010.S33222.l001.L001.fttf (continuation file)

6.3.1.3 Evaluating trace files with *fttrace*

Note on usage

Function: Evaluating trace files

User group: FT user and FT administrator

Functional description

Trace files for all protocols (openFT, FTAM and ftp protocol) are evaluated with the *fttrace* command.

Format

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

Description

-h

Outputs the command syntax on screen. Any specifications after *-h* are ignored.

-d

Specifies that the trace files are to be output in hexadecimal format (dump format). However, this does not function with the FTP protocol.

If you do not specify *-d* then the files are output in printable form, default value.

-sl =n | -sl =l | -sl =m | -sl =h

Specifies the security level for the output if the files are output in printable format (also see the note):

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 if *-s/* is not specified.

h (high)

Passwords, user IDs, transfer admissions, account numbers, follow-up processing commands and file names are overwritten with XXX.

i If the files are output in dump format (*-d*) then, irrespective of the value specified in *-s/* , the lowest security level (*-s/=n*) is always used since the trace files are output without any further interpretation or evaluation and may therefore also contain user IDs and passwords in clear text.

-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 start of the file.

-t =hh:mm:ss

(to)

Specifies the time up to which trace entries in the trace file are to be evaluated. You enter the time in the format hours:minutes:seconds (2 digits each).

If you do not specify an end time then trace entries are output up to the end of the file.

tracefiles

Name(s) of the trace file(s) that you want to evaluate. You can specify multiple trace files and wildcards can be used.

6.4 Additional diagnostic information

- Displaying diagnostic records
- Message file for console commands
- Output diagnosis information with diaginfo (Windows systems)

6.4.1 Displaying diagnostic records

Unlike trace files, diagnostic records are written only if an error occurs. You can output these diagnostic records with the *ftshwd* command.

You can output the diagnostic records in the openFT Explorer using the *Show Diagnosis Information* command in the *Administration* menu.

6.4.2 Message file for console commands

Unix systems

Console outputs are sent to the Unix console. To keep track of these over extended periods, the console outputs generated by openFT are also written to the *conslog* file. *conslog* is located in the *log* directory of the openFT instance. In the case of the standard instance the path name is */var/openFT/std/log/conslog*.

You can output the messages in the openFT Explorer using the *Show Console Messages* command in the *Administration* menu.

Windows systems

In order to use the diagnostic trace information in console output, the output is also stored in the file *conslog*. *conslog* is located in the *log* directory of the openFT instance.

You can output the messages in the openFT Explorer using the *Show Console Messages* command in the *Administration* menu.

6.4.3 Output diagnosis information with diaginfo (Windows systems)

The *diaginfo* command allows you to create further diagnostic information. To do this, start *diaginfo* with the *-a* option and redirect output to a file.

Example: `diaginfo -a > diag.txt`

You can then make this diagnostics file available to the Customer Service team.

6.5 Code tables

The code tables are useful to diagnose code conversion errors.

6.5.1 Code table EBCDIC.DF.04

		upper half byte															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
lower half byte	0					SP	&	-	ø	Ø	°	μ	ϕ	ù		Ù	0
	1				NBSP	é	/	É	a	j	—	£	A	J	+	1	
	2				â	ê	Â	Ê	b	k	s	¥	B	K	S	2	
	3				ä	ë	Ä	Ë	c	l	t	•	C	L	T	3	
	4				à	è	À	È	d	m	u	©	D	M	U	4	
	5				á	í	Á	Í	e	n	v	§	E	N	V	5	
	6				ã	ï	Ã	Ï	f	o	w	¶	F	O	W	6	
	7				å	ï	Å	Ï	g	p	x	¼	G	P	X	7	
	8				ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8	
	9				ñ	ß	Ñ	”	i	r	z	¾	I	R	Z	9	
	A				·	!	^	:	«	ª	¡	¬	SHY	1	2	3	
	B				.	\$,	#	»	º	¿	[ô	û	Ô	{	
	C				<	*	%	@	ð	æ	Ð	\	ö	ü	Ö	Ü	
	D				()	_	'	ý	¸	Ý]	ò	ù	Ò	}	
	E				+	;	>	=	þ	Æ	þ	´	ó	ú	Ó	Ú	
	F						?	"	±	¤	®	×	õ	ÿ	Õ	~	

Code table EBCDIC.DF.04 (character set corresponding to ISO-8859-1)

6.5.2 Code table ISO 8859-1

		upper half byte															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
lower half byte	0			SP	0	@	P	`	p			NBSP	°	À	Ð	à	ð
	1			!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
	2			"	2	B	R	b	r			¢	²	Â	Ò	â	ò
	3			#	3	C	S	c	s			£	³	Ã	Ó	ã	ó
	4			\$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô
	5			%	5	E	U	e	u			¥	µ	Å	Ö	å	ö
	6			&	6	F	V	f	v				¶	Æ	Ø	æ	ø
	7			'	7	G	W	g	w			§	·	Ç	×	ç	÷
	8			(8	H	X	h	x			¨	¸	È	Ø	è	ø
	9)	9	I	Y	i	y			©	¹	É	Ù	é	ù
	A			*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú
	B			+	;	K	[k	{			«	»	Ë	Û	ë	û
	C			,	<	L	\	l				¬	¼	Ì	Ü	ì	ü
	D			-	=	M]	m	}			SHY	½	Í	Ý	í	ý
	E			.	>	N	^	n	~			®	¾	Î	Þ	î	þ
	F			/	?	O	_	o				¯	¿	Ï	ß	ï	ÿ

Code table ISO 8859-1

7 Appendix

This chapter contains the following information:

- Important CMX commands (Unix systems)
- Entering transport system applications in the TNS
- openFT in a cluster with Unix based systems
- The openFT instance concept in a Windows cluster
- FarSync X.25 transport system under Linux and Windows systems
- Sample files

7.1 Important CMX commands (Unix systems)

This section contains a short description of the most important CMX commands needed for the openFT configuration when openFT is used with CMX. You will find detailed information in the manual „CMX Operation and Administration“.

7.1.1 *tnsxc*om - Create the TS directory

With the *tnsxc*om command you can transfer files in the *tnsxfrm* format to TS directories. You can set different modes for functions such as the syntax check, update or recreating the TS directory.

The command has the following syntax (abbreviated):

***tnsxc*om [-l -s -S -u -i] [file]**

The options have the following meanings:

- l** LOAD mode
*tnsxc*om takes the entries from the file *file* one at a time and fills the (previously empty) TS directory with the syntactically correct entries.
 - s** CHECK mode
*tnsxc*om only applies the syntax check to the file *file* and records any possible syntax errors. The TS directory is not changed.
 - S** CHECK-UPD mode
Like for the *-s* option, the syntax check is run on the entire file *file* in the first run. If no syntax errors are found, then *tnsxc*om updates the TS directory in a second run.
 - u** UPDATE mode
*tnsxc*om takes the entries from the file *file* one at a time and merges the syntactically correct entries in the TS directory. Missing entries are created and existing entries are updated during this process.
 - i** INTERACTIVE mode
*tnsxc*om reads entries in the *tnsxfrm* format from stdin after it has indicated it is ready to receive input by outputting a prompt and merges them in the TS directory. Missing entries are created and existing entries are updated during this process.
- file** The name of the file with the entries in the *tnsxfrm* format that are to be evaluated when the *-l*, *-s*, *-S* or *-u* options are specified. You can specify more than one file.

Examples

- The following call transfers the entries in the file *input.dir* to the current TS directory:

```
tnsxc om -S input.dir
```
- You want to delete the \$FJAM entry from the TS directory. For this to be possible, the input file *upd.dir* must contain the following entry:

```
$FJAM DEL
```

The call is as follows: `tnsxc om -u upd.dir`

7.1.2 *tnsxprop* - Output properties of TS applications

tnsxprop outputs all values of all properties that are located in a TS directory for the specified TS applications to stdout in a printable format.

You can specify in which format the properties are to be output using the first parameter.

The TS applications are determined by the parameter values for *name*. The parameter values for *name* can also be passed to *tnsxprop* from *file*. If no data was specified for *name* or *file*, then *tnsxprop* prepares the properties of all TS applications in the TS directory in the specified format.

The command has the following syntax (abbreviated):

tnsxprop [-S | -h] [-f file] [name ...]

- S This is the default setting. This option can be used to output the properties in symbolic form in the *tnsxfrm* format.
- h This option can be used to prepare the properties in hexadecimal form. The output is a string of hexadecimal digits together with the corresponding bit representation in which the lowest valued bit is located on the far right.
- f file You specify for *file* the name of a file that contains the GLOBAL NAMES of the TS application whose properties are to be queried. The GLOBAL NAMES are to be specified as described under *name*.

name The GLOBAL NAME of the TS application in the TS directory is to be specified as follows for *name*:

NP5.NP4.NP3.NP2.NP1

The individual NP*i*'s are the name attributes of the GLOBAL NAME.

NP5 is name attribute [5], i.e. it is the part of the name of the lowest hierarchy level. NP1 is name attribute [1], i.e. it is the part of the name of the highest hierarchy level. The name attributes are to be specified in ascending order hierarchically from left to right.

If one of the name attributes for a GLOBAL NAME does not contain data (e.g. NP4) and a name attribute of a higher level follows this name attribute (e.g. NP3), then only the separator (.) is to be specified for the name attribute that does not contain data. A series of separators at the end of the value of *name* does not have to be specified.

If the name attributes contain special characters whose special meaning would cause the syntax to take on multiple meanings, then these special characters must be delimited using the backslash (\). When in doubt, you should delimit every special character. Superfluous characters are ignored by *tnsxprop*.

If you specify an asterisk (*) for a name attribute, then *tnsxprop* returns the properties of all TS applications that match all other name attributes specified in *name* (TS_RE-STRICED filter mode).

Examples

1. The properties of the TS application that only has name attribute [5] set to the value *example_1* are to be output in hexadecimal form:

```
tnsxprop -h example_1
```

2. The properties of the TS application that only has name attribute [5] set to the value *example_1* are to be output in symbolic form:

```
tnsxprop example_1
```

3. The properties of all TS applications are to be output to a file *tns*:

```
tnsxprop > tns
```

7.2 Entering transport system applications in the TNS

It is not necessary to use the Transport Name Service (TNS) for linking over TCP/IP. If you nevertheless use the TNS; for instance if you link to transport systems other than TCP/IP or you wish to make use of existing TNS entries, then CMX must be installed and operation with CMX and TNS must have been explicitly activated in the operating parameters, e.g. using the command `ftmodo -tns=y -cmx=y`. Alternatively, in the openFT Explorer you can open the *Administration* menu and choose the *Operating Parameters* command then go to the *Protocols* tab and enable the options *Use TNS* and *Use CMX*.

The TNS identifies a transport system application (TS application) by means of a symbolic name known as the GLOBAL NAME. The symbolic name generally consists of up to five name parts.

These symbolic names are assigned address information. The necessary specifications, such as station name, application name, port number, etc. can be obtained from your network administrator.

Depending on the installation variant, (new installation, update installation) and the type of link, certain entries are made during the installation of openFT provided that CMX was installed on the system before the installation of openFT (see also [section "TNS entries created automatically"](#).)

Creating default TNS entries via a script

If CMX is not installed until after openFT or if there are no current TNS entries for openFT then you can create the default TNS entries for openFT as follows:

Unix systems:

Call the script `/opt/openFT/bin/ftbin/ftgentns`.

Windows systems:

Call the program `createtns.exe`. `createtns.exe` is located in the `\bin\ftbin` subdirectory of the openFT installation directory.

Creating TNS entries manually

Unix systems:

The entries in the TNS can be made with the aid of the TNS compilers `tnsxcnm`. To do this, enter the TS applications in a file, and then translate this file with the aid of the TNS compilers `tnsxcnm` (see [section "tnsxcnm - Create the TS directory"](#)).

Some Unix systems also provide a graphical user interface (menu system or Web interface) that you can use to enter the partner systems. For further details, refer to the CMX manual.

Windows systems:

You create TNS entries using the graphical user interface *TNS User Interface* that can be called from the Start menu (*Start - Programs - PCMX-32 - TNS User Interface*).

It can also be useful to enter the remote TS applications of the partner systems which are to issue requests to the local system. Ensure that the name, by which requests are processed with this partner, correspond to the instance ID of the remote system. If there is any doubt, a TNS input is required.

In this case, In the case of WAN partners, the partner is easier to identify for requests issued in the remote system. For example, the name of the partner as entered in the TNS is recorded in the log records. With FTAM partners which are not interconnected via TCP/IP, an entry in the TNS is the precondition.

Which entries are created or modified for which installation variant and which type of link are explained in the following section entitled “TNS entries created automatically”.

The procedure for the entry of remote TS applications is explained starting in [section “Definition of a remote TS application for openFT”](#).

TNS entries for cluster configurations

i When using cluster with openFT it is recommended to do use TNS and CMX.

Please note that cluster configurations are only supported for TCP/IP. You will therefore need to check all openFT-specific TNS entries for cluster configurations and delete those transport system entries that are not related to TCP/IP. (i.e. everything except for RFC1006 and LANINET). You will find some examples of this in [section “Sample”](#) (Windows systems) or in [section “Example 1: one fail-safe instance”](#) and in [section “Example 2: Fail-safe capability for both computers in the cluster”](#) (Unix systems).

7.2.1 TNS entries created automatically

If CMX is installed on the system then, depending on the installation variant, when openFT is installed, certain FT applications are automatically entered in the TNS or the existing entries are modified.

It is generally advisable not to modify the applications entered during the installation. If this is required in any case, it must be ensured that the port number of the \$FJAM entry is divisible by 100 and that the port number of the \$FJAMOUT entry is equal to the port number of the \$FJAM entry + 1. If your system is protected by a firewall and is to be accessible from the outside, the \$FJAM input port must be released in the firewall.

TNS entries for a new installation

Depending on the platform, a maximum of the following entries are made:

Unix systems (see also the file /opt/openFT/config/tnsstd)

```
$FJAM\  
  TSEL      WANNEA T'$FJAM'  
  TSEL      LANSBKA T'$FJAM'  
  TSEL      WANSBKA T'$FJAM'  
  TSEL      OSITYPE T'$FJAM'  
  TSEL      RFC1006 T'$FJAM'  
  TSEL      LANINET A'1100'  
$FJAMOUT\  
  TSEL      WANNEA T'$FJAMOUT'  
  TSEL      LANSBKA T'$FJAMOUT'  
  TSEL      WANSBKA T'$FJAMOUT'  
  TSEL      OSITYPE T'$FJAMOUT'  
  TSEL      RFC1006 T'$FJAMOUT'  
  TSEL      LANINET A'1101'  
$FTAM\  
  PSEL      V''  
  SSEL      V''  
  TSEL      LANSBKA T'$FTAM'  
  TSEL      WANSBKA T'$FTAM'  
  TSEL      OSITYPE T'$FTAM'  
  TSEL      RFC1006 T'$FTAM'  
  TSEL      LANINET A'4800'
```

Windows systems (layout after exporting in a text file)

```

$FJAM\
    TSEL          OSITYPE T' $FJAM'
    TSEL          WANSBKA T' $FJAM'
    TSEL          WANFAR  T' $FJAM'
    TSEL          RFC1006 T' $FJAM'
    TSEL          LANINET A' 1100'
    APPTYPE       openFT
$FJAMOUT\
    TSEL          OSITYPE T' $FJAMOUT'
    TSEL          WANSBKA T' $FJAMOUT'
    TSEL          WANFAR  T' $FJAMOUT'
    TSEL          RFC1006 T' $FJAMOUT'
    TSEL          LANINET A' 1101'
    APPTYPE       openFT
$FTAM\
    TSEL          OSITYPE T' $FTAM'
    TSEL          WANSBKA T' $FTAM'
    TSEL          WANFAR  T' $FTAM'
    TSEL          RFC1006 T' $FTAM'
    TSEL          LANINET A' 4800'
    SSEL          V' '
    PSEL          V' '
    APPTYPE       openFT
TranSON\
    TA            RFC1006 127.0.0.1 PORT 4444 A' SOCKS4'
    APPTYPE       PROXY

```

*) The TranSON entry is created by PCMX-32.

The local TS application \$FJAM is the contact for inbound requests from openFT partners, \$FJAMOUT for outbound requests to openFT partners.

The local TS application \$FTAM is the contact for all inbound and outbound requests with FTAM partners.

i *Windows systems:* As of openFT V11, the transport selector for the \$FTAM application was changed from SNI-FTAM to \$FTAM.

TNS entries for an update installation

The following applies with an update installation:

- At most, those TNS entries are created that are also created with a new installation.
- If entries of the form \$FJAM_OUTBOUND, *ftstd* or *ftstdisdn* are present, they are deleted.
- All existing entries other than \$FJAM_OUTBOUND, *ftstd* or *ftstdisdn* are retained unchanged.

7.2.2 Definition of the local TS application for openFT-FTAM

If you wish to use openFT-FTAM during operation with TNS, the local application \$FTAM must be defined. This is done automatically during new installation or full installation and update installation if CMX is installed and if no \$FTAM entry is present. The local application \$FTAM is used for all request with FTAM partners (outbound and inbound).

Special points

With the TCP/IP-LAN transport system, two entries must be made for the symbolic name:

- an RFC1006 entry with the transport selector. Enter the relevant symbolic name \$FTAM as transport selector. The entry must be made TRANSDATA format (indicator T).
- a LANINET entry with the port number. The port number is specified in ASCII format.

Unix systems:

You must make the entry in a defined format (see samples).

More details on this topic can be found in the CMX manual.

The GLOBAL NAME \$FTAM is fixed. T'\$FTAM' is recommended for the transport selector. The entries PSEL V'' and SSEL V'' are absolutely necessary.

Windows systems:

You make the entries via the *TNS User interface* GUI.

More details on this topic can be found in the online help of the *TNS User interface*.

The GLOBAL NAME \$FTAM is fixed. T'\$FTAM' is recommended for the transport selector. You can choose any presentation selector and session selector you want. You are recommended to choose the empty format unless the partners require another specification.

Sample entries for openFT-FTAM on Sparc Solaris

```
$FTAM\  
PSEL    V''          ; empty presentation  
SSEL    V''          ; empty session selector  
TSEL    WANSBKA T'$FTAM' ; entry for WAN-CONS, ISDN-CONS  
TSEL    LANSBKA T'$FTAM' ; entry for ETHN-CLNS/passive  
                ; necessary for link to CMX V3.0  
TSEL    OSITYPE T'$FTAM' ; entry for ETHN-CLNS/active  
TSEL    RFC1006 T'$FTAM' ; entry for TCP/IP-RFC1006  
TSEL    LANINET A`4800`  ; entry for TCP/IP
```

7.2.3 Definition of a remote TS application for openFT

You must ensure that the name, by which requests are processed with this partner, correspond to the instance ID of the remote system. If there is any doubt, a TNS input, whose global name is the instance ID, is needed.

For each further partner system which is to be accessible for requests issued locally, it is necessary to make a TNS entry. In both of the cases described above, additional TNS entries must be made for the partner systems, and separate names assigned to the partner systems. The entries are made

- in the TNS User Interface (Windows systems)
- in the file translated using the TNS compiler *tnsxcom* or the graphical interface (Unix systems)

As symbolic name (GLOBAL NAME), you must use an alphanumeric name containing up to 78 characters. No special characters may be used, except for:

- “.” as separator
- “#” . The entry behind the hash “#” is used to differentiate entries with the same prefix. In this way, it is possible to enter a partner (who has several addresses) several times with the same name (prefix). This is only useful for inbound requests. Here, the partner system is always displayed with the same partner address (corresponding to the prefix).

You are free to select the symbolic name. However, it must be unique in the local system. The further entries to be made depends on the how the remote system is connected to the network. The entries must be made in TRANSDATA format (indicator *T*). You can obtain the information required to make the entries from the network administrator.

7.2.3.1 Sample entries for openFT partners (Unix systems)

The examples listed below assume that the corresponding transport system is installed on your Unix computer.

i Note that only TCP/IP-RFC1006 is present by default on Unix systems.

- Entry of a partner address (for a openFT (BS2000) partner) for transfer via TCP/IP-RFC1006 (Port 102):

```
ftbs2\  
      TA      RFC1006 123.4.5.68      T'$FJAM'  
;                Internet addr. T selector
```

- Entry of a PCMX partner address for transfer via TCP/IP-RFC1006 and a CMX or Windows partner:

```
ftrfc\  
      TA      RFC1006 123.4.5.67      PORT 1100 T'$FJAM'  
;                Internet addr. Portno      T selector
```

- Entry of variable Internet addresses for one and the same partner with the name *mobile* (e.g. a Notebook used from different locations and thus connected via different Internet addresses):

```
mobile\  
      TA      RFC1006 100.22.33.45      PORT 1100 T'$FJAM'  
;                Internet-addr1.  Portno      T selector  
mobile#1\  
      TA      RFC1006 101.20.30.40      PORT 1100 T'$FJAM'  
;                Internet addr2.  Portno      T selector  
mobile#2\  
      TA      RFC1006 102.21.31.41      PORT 1100 T'$FJAM'  
;                Internet-addr3.  Portno      T selector
```

7.2.3.2 Sample entries for openFT partners (Windows systems)

The following examples are created using the *TNS User Interface*. Values in *italics* may be changed.

- Partner address entry for openFT (Windows) for transfer via TCP/IP RFC100:

```
Global Name      ftwin
Application type openFT
P selector       <none>
S selector       <none>
T selector       TRANSDATA   $FJAM
Port number      1100
Proxy            <none>
System name      transport-system-dependent
```

- Partner address entry for openFT (BS2000) for transfer via TCP/IP RFC100:

```
Global Name      ftbs2
Application type openFT
P selector       <none>
S selector       <none>
T selector       TRANSDATA   $FJAM
Port number      102
Proxy            <none>
System name      transport-system-dependent
```

- Partner address entry for transfer via TranSON:

```
Global Name      fttranson
Application type openFT
P selector       <none>
S selector       <none>
T selector       TRANSDATA   $FJAM
Port number      1100
Proxy            TranSON
System name      transport-system-dependent
```

The global application TranSON with the proxy address is registered during the installation of PCMX-32.

- Partner address entry for transfer via X.25:

Global Name *ftiso*
Application type openFT
P selector <none>
S selector <none>
T selector TRANSDATA \$FJAM
Transport protocol
information <empty> 1100
Subnetwork *X.121*
Transport protocol
class class2 (0 is possible)
X.25-DTE address *transport-system-dependent*

7.2.4 Definition of remote TS applications for openFT-FTAM

i In order to use the Transport Name Service, CMX must be installed and operation with CMX and TNS must have been explicitly activated in the operating parameters, e.g. using the command `ftmodo -tns=y -cmx=y`. Alternatively, in the openFT Explorer you can open the *Administration* menu and choose the *Operating Parameters* command and then go to the *Protocols* tab and enable the options *Use TNS* and *Use CMX*

In the case of all partner systems that can be accessed via TCP/IP or X.25, no TNS entries are required since you can specify the partner address directly or enter it in the partner list.

The presentation/session and transport selector entries can be made in ASCII (A'...'), EBCDIC (E'...'), TRANSDATA format (T'...') or hexadecimal (X'...'). Presentation and session selectors may only be between 0 and 16 bytes long. If the presentation or session selector is missing, on Unix systems the entries `PSEL V' '` or `SSEL V' '` are absolutely necessary and on Windows systems it is essential to specify *Empty format*. On Unix systems with transport addresses for FTAM partners, no CC list may be specified.

If a partner has different addresses for inbound and outbound requests, to simplify administration you can define a dummy entry containing the incoming sender address for the inbound side. To do this you enter a "#" (hash), followed by a number in part 5 of the global name.

Special points (Unix systems)

The entries of the file to be translated with `tnsxcnm` must in principle look the same as in the following examples in [section "Sample entries for FTAM partners \(Unix systems\)"](#). You can use the following checklist to assist you.

Checklist

The following checklist is intended to help you gather the data required for the TNS entry of an FTAM partner. The questions must be answered by the FTAM partner.

- openFT-FTAM sets up the connection.

Which values do the following parameter have (with specification of coding):

a)	called LAN address	_____		
b)	called TSEL	_____	Code:	_____
c)	called SSEL	_____	Code:	_____
d)	called PSEL	_____	Code:	_____
e)	Protocol Identifier (Layer 3 CUD)	_____		
f)	called APT	_no _____NILAPTitle __ ¹⁾		
g)	called AEQ	_no _____ ¹⁾		
h)	calling APT	_no _____NILAPTitle __ ¹⁾		

¹⁾APT (Application Process Title) and AEQ (Application Entity Qualifier) are not specified in the TNS entries, but in the openFT commands. Some FTAM partners expect APTs and possibly AEQs; others expect no APTs/AEQs to be specified.

2. The partner sets up the connection.

Which values do the following parameters have (with specification of coding):

a)	calling LAN address	_____		
b)	calling TSEL	_____	Code:	_____
c)	calling SSEL	_____	Code:	_____
d)	calling PSEL	_____	Code:	_____

You must observe correct notation (uppercase and lowercase) and remember that blanks and X'00' must be specified correctly for selectors.

7.2.4.1 Sample entries for FTAM partners (Unix systems)

The examples listed below assume that the corresponding transport system is installed on your Unix computer.

- Entry of a partner address for transfer via TCP/IP-RFC1006. The partner supports the standardized port number 102 of RFC1006.

```
ftamrfc\  
  PSEL    V''  
  SSEL    V''  
  TA      RFC1006 123.4.5.67      T'$FTAM'  
;                          Internet addr. T selector
```

- Entry of a partner address (openFT (Windows)

<= V10.0 with FTAM functionality) for transfer via TCP/IP-RFC1006 (Port 4800) :

```
ftamwnt\  
  PSEL    V''  
  SSEL    V''  
  TA      RFC1006 123.4.5.68      PORT 4800    A'SNI-FTAM'  
;                          Internet addr Portno    T selector
```

7.2.4.2 Sample entries for FTAM partners (Windows systems)

Values in *italics* may be changed.

- FTAM partner address entry for openFT (Windows) as of V11.0 for transfer via TCP/IP RFC1006:

Global Name	<i>ftamwin</i>
P selector	EMPTY FORMAT
S selector	EMPTY FORMAT
T selector	TRANSDATA \$FTAM
Port number	<i>4800</i>
Proxy	<none>
System name	<i>transport-system-dependent</i>

- FTAM partner address entry for openFT (Windows) < V11.0 for transfer via TCP/IP RFC1006:

Global Name	<i>ftamwina</i>
P selector	EMPTY FORMAT
S selector	EMPTY FORMAT
T selector	ASCII SNI-FTAM
Port number	<i>4800</i>
Proxy	<none>
System name	<i>transport-system-dependent</i>

- FTAM partner address entry for openFT (Unix systems) for transfer via TCP/IP RFC1006 if CMX or PCMX as of V4.0 is used on the Unix system:

Global Name	<i>ftamunix</i>
P selector	EMPTY FORMAT
S selector	EMPTY FORMAT
T selector	TRANSDATA \$FTAM
Port number	<i>4800</i>
Proxy	<none>
System name	<i>transport-system-dependent</i>

- Partner address entry for openFT-FTAM (BS2000) for transfer via TCP/IP RFC1006:

Global Name	<i>ftambs2</i>
P selector	EMPTY FORMAT
S selector	EMPTY FORMAT
T selector	TRANSDATA \$FTAM
Port number	<i>102</i>
Proxy	<none>
System name	<i>transport-system-dependent</i>

- Partner address entry for transfer via TranSON:

```

Global Name      ftamtranson
P selector       EMPTY FORMAT
S selector       EMPTY FORMAT
T selector       TRANSDATA   $FTAM
Port number      102
Proxy            TranSON
System name      transport-system-dependent

```

The global application TranSON with the proxy address is registered during the installation of PCMX-32.

- Example of interconnection with DEX system

The partner requires the selectors in ASCII format, but itself sends empty selectors in its sender address if it has the initiative.

```

Global Name      dex
Part 5
P selector       ASCII      TS
S selector       ASCII      TS-SSAP
T selector       ASCII      TS-TSAPEAF
System name      transport-system-dependent

```

The following entry is required if the initiative comes from the DEX system. Its sole purpose is to identify an initiator.

```

Global Name      dex#01
Part 5
P selector       <none>
S selector       <none>
T selector       ASCII      TS-TSAPEAF
System name      transport-system-dependent s

```

7.3 openFT in a cluster with Unix based systems

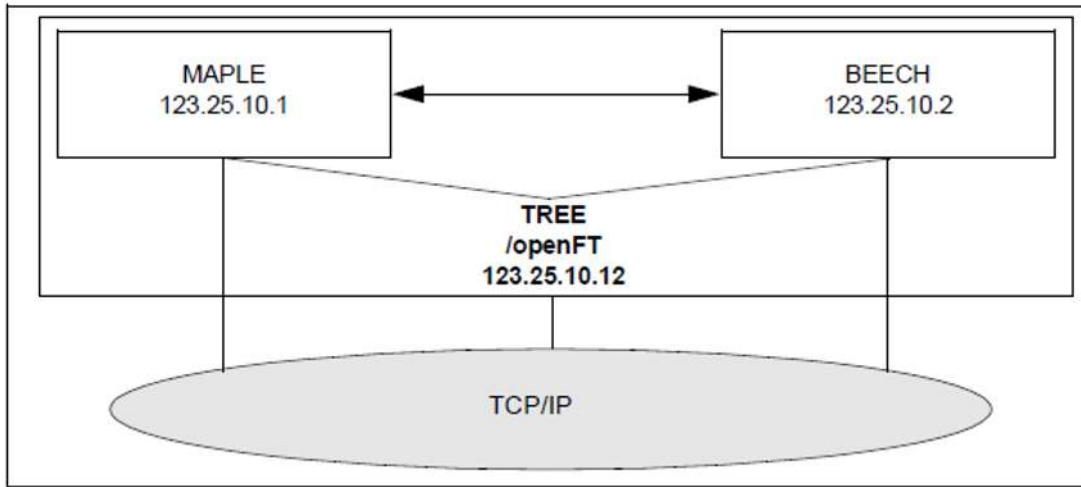
Software requirements

The same version of openFT must be installed on all nodes of the cluster. If you are using the TNS during operation with CMX please refer to [section “Notes for using TNS”](#).

You are recommended to work without CMX and TNS.

7.3.1 Example 1: one fail-safe instance

The cluster TREE (Unix based systems, IP-address 123.25.10.12) consists of the two computers MAPLE (IP-address 123.25.10.1) and BEECH (IP-address 123.25.10.2). The failure management concept allows TREE to run either on MAPLE or BEECH. Only one openFT instance is fail-safe in this case.



openFT in a cluster: one fail-safe instance

Configure the cluster in such a way that a disk is always available. In this example, it is the directory */openFT*.

Required steps for the computer MAPLE

1. Install openFT (including the add-on products openFT-CR, openFT-FTAM and openFT-FTP, if required)
2. Deactivate openFT:

```
ftstop
```

3. If you are working with CMX and TNS, you must adapt the \$FJAM, \$FJAMOUT and \$FTAM (if required) TNS inputs to the system. They may only contain RFC1006 and LANINET inputs, see above.
4. Set the address for the instance *std*:

```
ftmodi std -addr=MAPLE
```

The instance *std* logs in exclusively at the address MAPLE. All other addresses on the computer are available for other instances.

5. Activate openFT on the instance *std* and set the ID, if this did occur automatically during installation:

```
. ftseti std
[ftmodo -id=MAPLE.FOREST.NET]
ftstart
```

6. Mount the disk */openFT* on MAPLE.
7. Create the new instance *cluster* and check it. The directory */openFT* must exist, whereas the directory */openFT/cluster* must not exist:

```
ftcrei cluster /openFT/cluster -addr=TREE.FOREST.NET
ftshwi @a -l
Instance Address          directory
-----
cluster  TREE.FOREST.NET /openFT/cluster
std      MAPLE          /var/openFT/std
```

8. If authentication is to be used in the instance *cluster*, then public keys from the partner systems must be stored in the directory */openFT/cluster/syskey*, or the public key from the directory */openFT/cluster/config* must be made available to the partner systems.

9. Deactivate the instance *cluster*:

```
ftseti std; ftdeli cluster
```

Required steps on for the computer BEECH

1. Install openFT (including the add-on products openFT-CR openFT-FTAM and openFT-FTP, if required)

2. Deactivate openFT:

```
ftstop
```

3. If you are working with CMX and TNS, you must adapt the \$FJAM, \$FJAMOUT and \$FTAM (if required) TNS inputs on the system if they exist. They may only contain RFC1006 and LANINET inputs, see above.

4. Set the address of the instance *std*:

```
ftmodi std -addr=BEECH
```

The instance *std* logs in exclusively at the address BEECH. All other addresses on the computer are available for other instances.

5. Activate openFT on instance *std* and set the ID, if this did not occur automatically during installation:

```
. ftseti std
[ftmodo -id=BEECH.FOREST.NET]
ftstart
```

6. Next, make a shell script for administering the instance that handles the Events *start*, *stop*, and *check*. The script must be available and properly configured on the computers **MAPLE** and **BEECH**. It might look like the following when RMS (Reliant Monitor Services) is used:

```

PAR=$1
BIN=/opt/bin; export BIN
INST=cluster
OPENFTINSTANCE=$INST; export OPENFTINSTANCE
case $PAR in
start) OPENFTINSTANCE=std; export OPENFTINSTANCE
    $BIN/ftcrei $INST /openFT/$INST
    case $? in
        0|5) continue;;
        *) exit 1;;
    esac
    OPENFTINSTANCE=$INST; export OPENFTINSTANCE
    $BIN/ftstart 2>/dev/null
    case $? in
        0|180) exit 0;;
        *) exit 1;;
    esac;;
stop) $BIN/ftstop 2>/dev/null
    case $? in

```

```

        0|181) continue;;
        *) exit 1;;
    esac
    OPENFTINSTANCE=std; export OPENFTINSTANCE
    $BIN/ftdeli cluster
    case $? in
        0) exit 0;;
        *) exit 1;;
    esac;;
check) VALUE=`$BIN/ftshwo -csv 2>/dev/null |fgrep FtStarted\
    |sed s/";"/" "/g`
    [ -z $VALUE ] && exit 1
    set $VALUE
    i=1
    FTROW=1
    while [ "$i" != "FtStarted" ]
    do shift
    FTROW=`expr $FTROW + 1`
    done
    FTSTART=`$BIN/ftshwo -csv|fgrep -v FtStarted|cut \
        -f$FTROW -d\;`
    if [ $FTSTART = '*NO' ]
    then # openFT server not active
    exit 1
    else # openFT server active
    exit 0
    fi
    ;;
esac

```

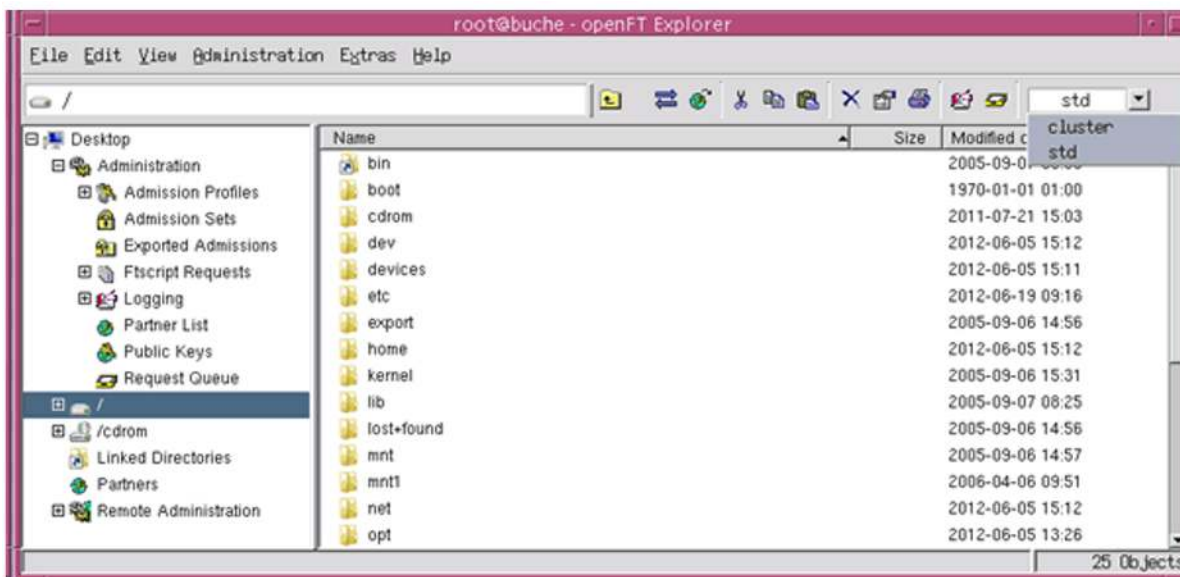
Working with individual instances

When everything is finished, there is a standard instance on both the MAPLE and BEECH computers which is not fail-safe. By making a selection on the openFT Explorer, or by executing the command `ftseti std`, you will be working with the respective standard instance. You can make use of all the openFT functions in the standard instances (e.g. set up admissions profiles, view log records, etc.). The standard instances on MAPLE and BEECH can be addressed normally from external systems using the addresses of these computers (123.25.10.1 or 123.25.10.2).

The fail-safe instance `cluster` is available on one of these two computers; the one on which the disk `/openFT` is currently mounted. You can work with the instance on this computer using the graphical user interface or by using the command

`.ftseti cluster` and use all of openFT functions available there. It is not necessary to know on

which computer the disk `/openFT` is mounted during this. You must choose TREE as the partner. The cluster TREE (openFT instance `cluster`) is addressed externally under the IP address 123.25.10.12.

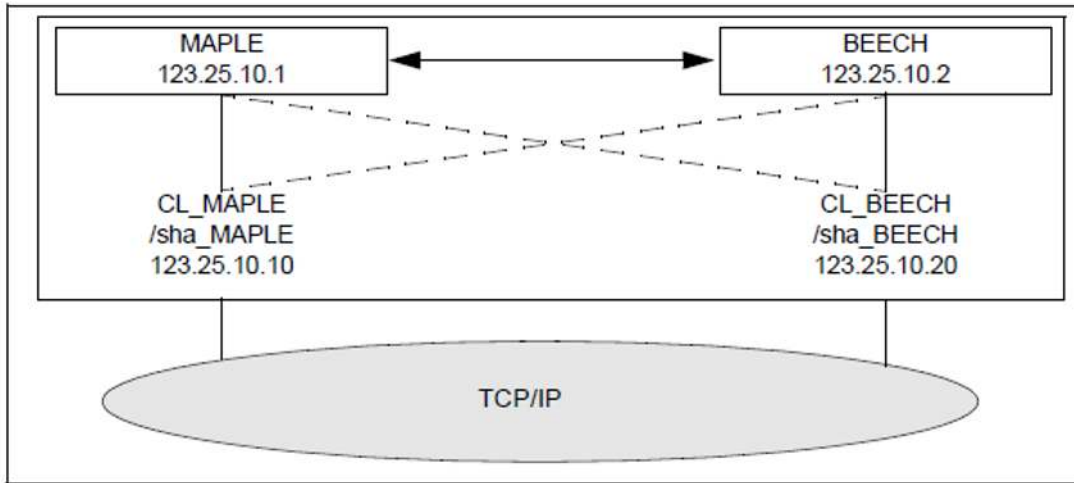


Selecting an instance in a cluster

7.3.2 Example 2: Fail-safe capability for both computers in the cluster

The cluster of Unix systems, once again, consists of two computers: MAPLE (IP address 123.25.10.1) and BEECH (IP address 123.25.10.2).

In this example, however, there is to be a fail-safe openFT instance available on each of the two computers. For this purpose, the computers are superimposed (MAPLE by CL_MAPLE (IP address 123.25.10.10) and BEECH by CL_BEECH (IP address 123.25.10.20). If the computer MAPLE fails, then CL_MAPLE is switched over to the computer BEECH. If the computer BEECH fails, then CL_BEECH is switched over to the computer MAPLE.



openFT in a cluster: fail-safe capability for both computers

Configure the cluster so that a disk is always available for each computer, for example: */sha_MAPLE* and */sha_BEECH*.

Required steps for the computer MAPLE

1. Configure a standard instance as shown in example 1.
2. Mount the disk */sha_MAPLE* and */sha_BEECH* on MAPLE.
3. Create and check the instances *MAPLE* and *BEECH*:

```
ftcrei MAPLE /sha_MAPLE/oFT -addr=CL_MAPLE.FOREST.NET
ftcrei BEECH /sha_BEECH/oFT -addr=CL_BEECH.FOREST.NET
ftshwi @a -l
Instance Address          Directory
-----
maple    CL_MAPLE.FOREST.NET    /sha_MEAPPLE/oFT
beech    CL_BEECH.FOREST.NET    /sha_BEECH/oFT
std      MAPLE                  /var/openFT/std
```

4. Deactivate the instances

MAPLE and *BEECH*:

```
ftdeli MAPLE
```

```
ftdeli BEECH
```

Required steps on the computer BEECH

1. Configure a standard instance as shown in example 1.
2. Next, make a shell script for controlling openFT on the computers MAPLE and BEECH that handles the events *start*, *stop*, and *check*. Both scripts must be available on both computers. When RMS is used, the shell script might look like the example below (in the script for BEECH, the name *MAPLE* must be substituted with *BEECH* in the following):

```
PAR=$1
BIN=/opt/bin; export BIN
INST=MAPLE
OPENFTINSTANCE=$INST; export OPENFTINSTANCE
case $PAR in
start) OPENFTINSTANCE=std; export OPENFTINSTANCE
    $BIN/ftcrei $INST /sha_MAPLE/oFT
    case $? in
        0|5) continue;;
        *) exit 1;;
    esac
    OPENFTINSTANCE=$INST; export OPENFTINSTANCE
    $BIN/ftstart 2>/dev/null
    case $? in
        0|180) exit 0;;
        *) exit 1;;
    esac;;
```

```

stop) $BIN/ftstop 2>/dev/null
  case $? in
    0|181) exit 0;;
    *) exit 1;;
  esac
OPENFTINSTANCE=std; export OPENFTINSTANCE
$BIN/ftdeli $INST
  case $? in
    0)exit 0;;
    1)exit 1;;
  esac;;
check) VALUE=`$BIN/ftshwo -csv|fgrep FtStarted \
  |sed s/";"/" "/g`
  set $VALUE
  i=1
  FTROW=1
  while [ "$1" != "FtStarted" ]
  do shift
    FTROW=`expr $FTROW + 1`
  done
  FTSTART=`$BIN/ftshwo -csv|fgrep -v FtStarted \
  |cut -f$FTROW -d\;`
  if [ $FTSTART = '*NO' ]
  then # openFT server not active
    exit 1
  else # openFT server active
    exit 0
  fi;;
esac

```

Working with the individual instances

When everything is finished, there is a standard instance on both the MAPLE and BEECH computers which is not fail-safe. By making a selection on the openFT Explorer, or by executing the command *ftseti std*, you will be working with the respective standard instance. You can make use of all the openFT functions in the standard instances (e.g. set up admissions profiles, view log records, etc.). The standard instances on MAPLE and BEECH can be addressed normally from external systems using the addresses of these computers (123.25.10.1 or 123.25.10.2).

The openFT instances *MAPLE* and *BEECH* are available on the computer, on which the corresponding disk is currently mounted. They can be used, as usual, via the openFT Explorer or the command interface.

In order to transfer files to these instances, the IP addresses of CL_MAPLE.FOREST.NET or CL_BEECH.FOREST.NET (123.25.10.10 or 123.25.10.20) can be addressed.

7.3.3 Notes for using TNS

On Solaris, TNS inputs are only allowed to contain TCP/IP components. An input file for the *tnsxcom* command could look like the following:

```
$FJAM      DEL
$FJAM\
  TSEL     RFC1006  T'$FJAM'   ; input for TCP/IP-RFC1006
  TSEL     LANINET  A'1100'    ; input for TCP/IP
$FJAMOUT   DEL
$FJAMOUT\
  TSEL     RFC1006  T'$FJAMOUT' ; input for TCP/IP-RFC1006
  TSEL     LANINET  A'1101'    ; input for TCP/IP
$FTAM      DEL
$FTAM
  PSEL     V' '      ; blank presentation selector
  SSEL     V' '      ; blank session selector
  TSEL     RFC1006  T'$FTAM'   ; input for TCP/IP-RFC1006
  TSEL     LANINET  A'4800'    ; input for TCP/IP
```

During this, the existing inputs in the TNS are overwritten by *tnsxcom*.

7.4 The openFT instance concept in a Windows cluster

Software Requirements

openFT (Windows) V12.1 (the same openFT version must be installed on all nodes of the cluster).

If you are using TNS then CMX must be installed and operation with CMX and TNS must have been explicitly activated in the operating parameters, e.g. using the command `ftmodo -tns=y -cmx=y`.

7.4.1 Sample

This is a Windows cluster *OPENFT* with the IP address 192.168.90.30 consisting of the two nodes *P870_DDM* (address 192.168.90.10) and *PN70_DDM* (address 192.168.90.20). the following applies:

- Each node contains a *std* instance
- There is one *cluster* instance for both cluster nodes, and this instance is assigned to one node at any one time, because it is located on a switchable cluster drive.

This means that there are three addressable openFTs on the two nodes of the cluster.

The failure concept lies in the fail-safe *cluster* instance (hostname OPENFT) that is online from the viewpoint of the cluster (either *P870_DDM* or *PN70_DDM*). If the *std* instance on the separate nodes is used, it must be noted that these are not failsafe.

Configure the Windows cluster in such a way that one disk is always available, which is managed by the cluster (Physical Disk e.g. S:\openFT).

7.4.1.1 Installation of openFT

Installation on the first node

Install openFT locally (plus the supplementary product openFT-CR if required):

- Always select a local disk for all paths (e.g. C:). Only activate the FTAM/FTP functionality if you possess the necessary license.
- Restart the computer. Enter the user password for openFT (*ftsetpwd* or openFT Explorer, menu *Administration*, *User password* command).
- Check if the identification is set properly (*ftshwo* or openFT Explorer) and correct it if necessary (*ftmodo -id=* or openFT Explorer).

Installation on the second node

Install openFT locally (plus the supplementary product openFT-CR if required), see the first node.

7.4.1.2 Configuration of resource-specific openFT properties of Cluster

Configure the cluster in such a way that one device, which contains the switching file of openFT, is always available (in this case S:\).

You will find an example for configuring the resource-specific openFT properties in [section “Configuring resource-specific openFT properties”](#).

7.4.1.3 Configuration of openFT

You are recommended to work without CMX and TNS (*ftmodo -cmx=n .tns=n*, default after new installation).

If using TNS make sure, that on both nodes of the cluster the same TNS entries are available, or use the registry replication of the cluster for this.

The asynchronous server is no separate service. The services *openFT* and *openFT Security Server* must always be started.

Configuration on first node (P870_DDM)

- Stop asynchronous openFT server: via *Administration - Stop Asynchronous Server* or use command *ftstop*.
- If you are using the TNS during operation with CMX then you must adapt the TNS entries \$FJAM, \$FJAMOUT and \$FTAM if required (only TCP/IP entries should be present).
- Set the address of the *std* instance:

```
ftmodi std -addr=P870_DDM
```

- On instance *std* start asynchronous openFT server: Choose instance *std* and select *Administration - Start Asynchronous Server* in the openFT Explorer or enter the command *ftstart*.
- Bring the first node online (using Move Group).
- Create a new instance *cluster* and check it (*OPENFT* is the hostname of the cluster, *OPENFT.XYZ.NET* the corresponding DNS-name; the directory *S:\openFT* must exist, the directory *S:\openFT\cluster* may not exist):

```
ftcrei cluster S:\openFT\cluster -addr=OPENFT.XYZ.NET
ftshwi @a -l
Instance  Address          Directory
-----  -
cluster   OPENFT.XYZ.NET    S:\openFT\cluster
std        P870_DDM          C:\ProgramData\Fujitsu Technology Solutions\
openFT\var\std
```

- Select the *cluster* instance in the drop down list of the openFT Explorer and start asynchronous openFT server automatically:
openFT Explorer *Administration - Operating Parameters*, activate the option *Start Asynchronous Server Automatically*.
Starting the service *openFT - cluster* will automatically start the asynchronous openFT server.
- Store the user password for the new instance via openFT Explorer or command *ftsetpwd*.
- If you use authentication on the *cluster* instance, public keys of partner systems must be stored in the directory *S:\openFT\cluster\syskey*, respectively make the public key of the directory *S:\openFT\cluster\config* available for partner systems.

Configuration on second node (PN70_DDM)

- Stop asynchronous openFT server (see above)
- If *Use TNS* is active: Adapt the TNS entries \$FJAM, \$FJAMOUT and \$FTAM if required (only TCP/IP entries should be present)
- Set address of *std* instance:

```
ftmodi std -addr=PN70_DDM
```

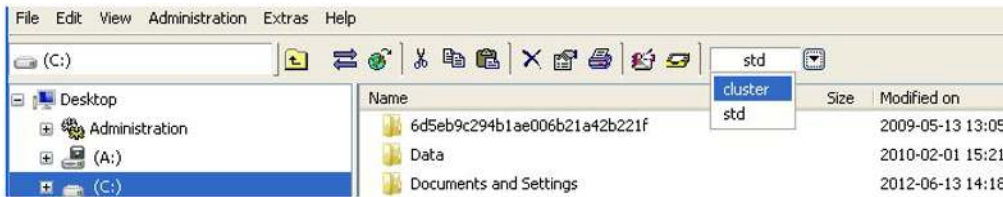
-
- On Instance *std* start asynchronous openFT server (see above)
 - Bring the second node online (using Move Group)
 - Activate and check instance *cluster* (you may not specify an address since the instance already exists):

```
ftcrei cluster S:\openFT\cluster
ftshwi @a -l
Instance Address          Directory
-----
cluster  OPENFT.XYZ.NET   S:\openFT\cluster
std      PN70_DDM              C:\ProgramData\Fujitsu Technology Solutions\
openFT\var\std
```

7.4.1.4 Operations with the individual openFT Instance

1st possibility: (openFT Explorer)

Set the instance in the drop-down list in the top right of the openFT Explorer.



2nd possibility: (command line)

- Cluster *OPENFT* (failsafe) at *P870_DDM* or *PN70_DDM* (depending on where the disk S:\ is online):

```
ftseti cluster
```

...any openFT command

```
ftcrep cluster1 FromOPENFT ...
```

```
ftshwl @a -nb=10
```

- Computer *P870_DDM* (not failsafe):

```
ftseti std
```

... any openFT command

```
ftcrep maple SendToP870_DDM ...
```

```
ftshwl -rc=@f
```

- Computer *PN70_DsDM* (not failsafe):

see computer *P870_DDM*

7.4.1.5 Use of the Windows cluster as an openFT Server

- In the case of transfers with the failsafe Windows cluster *OPENFT*, the host name *OPENFT* or the IP address 192.168.90.30 must be addressed, e.g.

```
ftshw OPENFT!. FromOPENFT -d
```

- In the case of transfers directly to the host *P870_DDM* (not failsafe), the host name *P870_DDM* or the IP address 192.168.90.10 must be addressed, e.g.

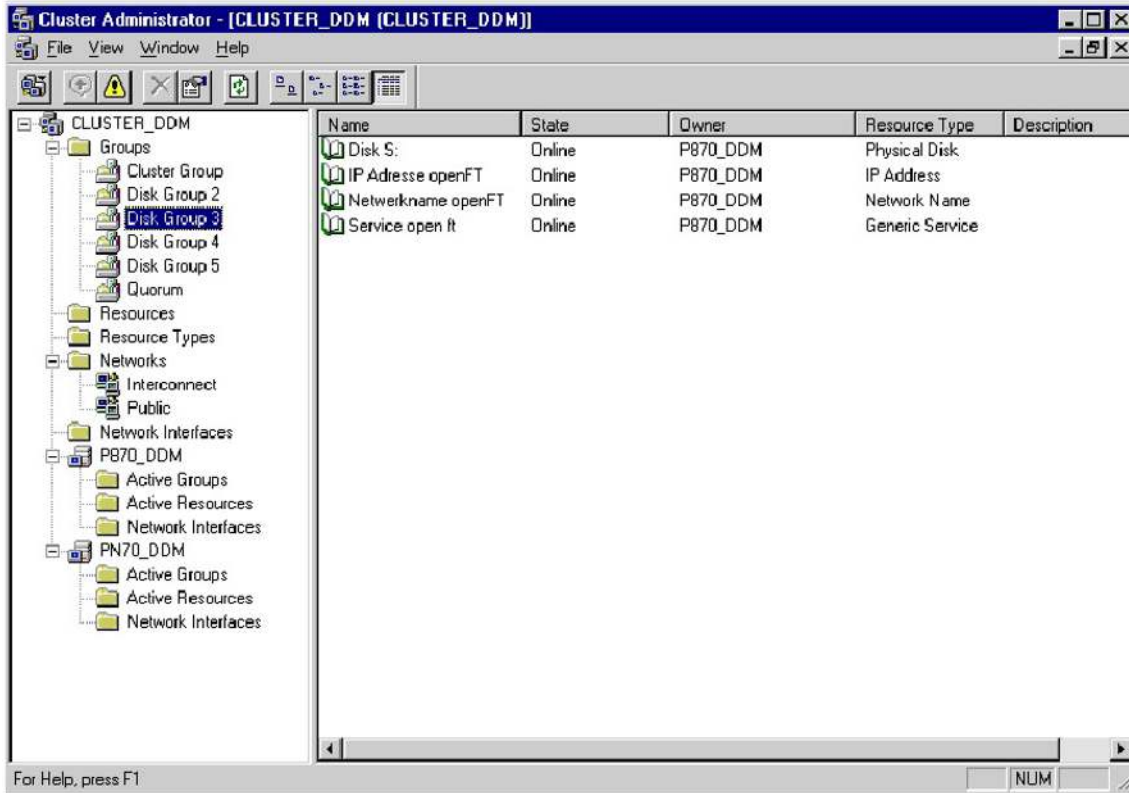
```
ncopy LocFile P870_DDM!RemFile SendToP870_DDM
```

- In the case of transfers directly to the host *PN70_DDM* (not failsafe), the host name *PN70_DDM* or the IP address 192.168.90.20 must be addressed, e.g.

```
ncopy PN70_DDM!RemFile LocFile GetFromPN70_DDM
```

7.4.2 Configuring resource-specific openFT properties

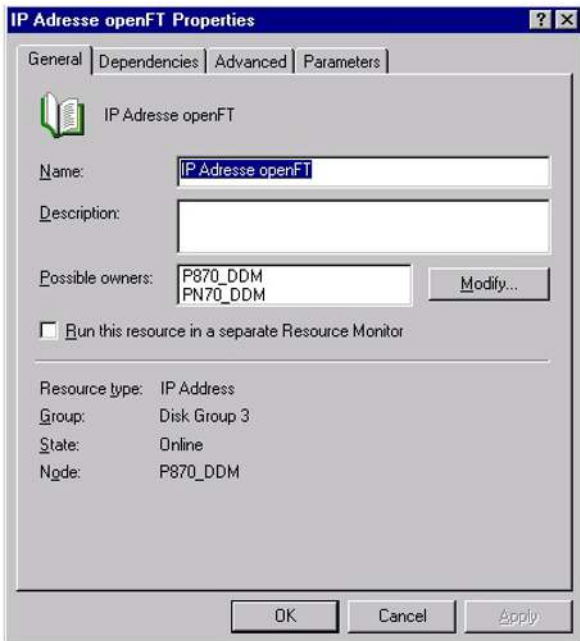
Use *Program – Administrative Tools – Cluster Configuration* in the Cluster Administrator Tool, which is a component of the Windows Server, to configure openFT in one of the two nodes of the Cluster.



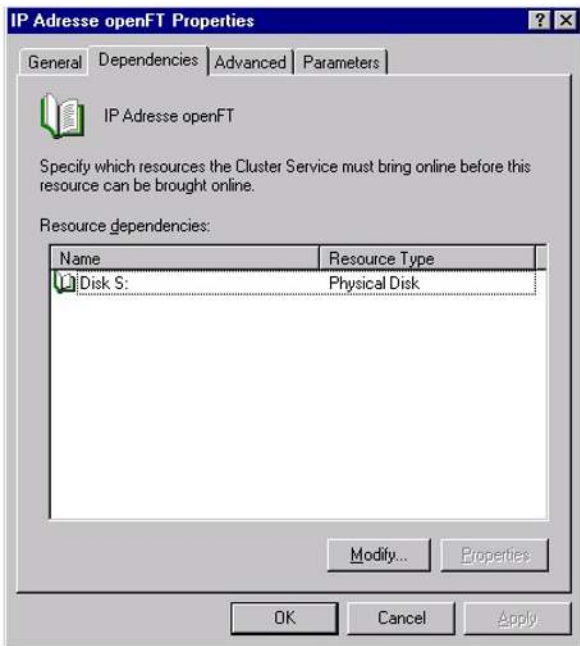
Set up the relevant resources with the following properties:

1.	Name	e.g. IP Address openFT
	Resource Type	IP Address
	Dependencies	Physical Disk (in this case Disk S:)
	Advanced	Use standard or customize
	Parameters	IP Address (namely the one openFT-Client needs to address the cluster)

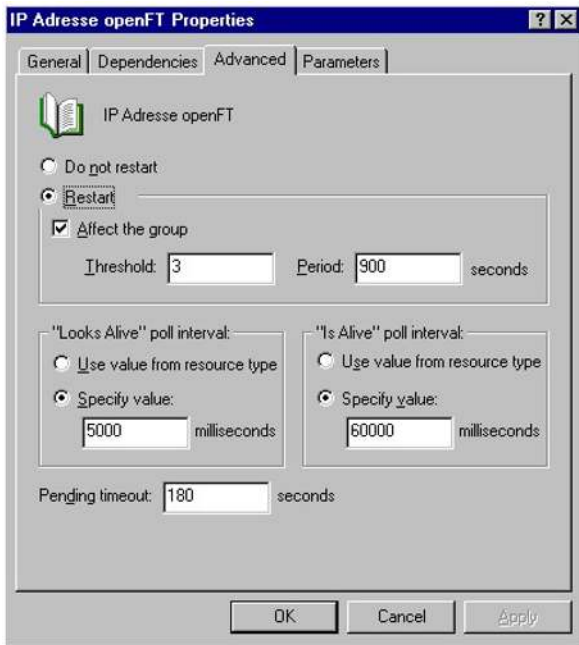
Press *Apply* to create the resource and bring it online (right mouse button – *Bring Online*).



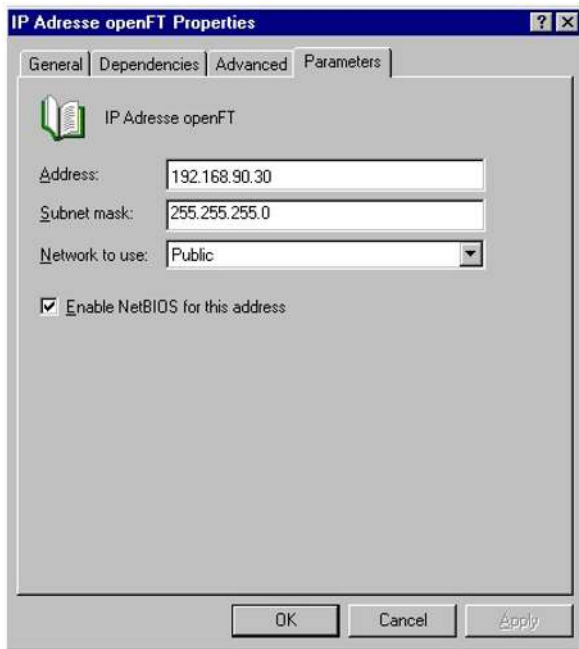
Configure openFT in a cluster: IP Address - General



Configure openFT in a cluster: IP Address - Dependencies



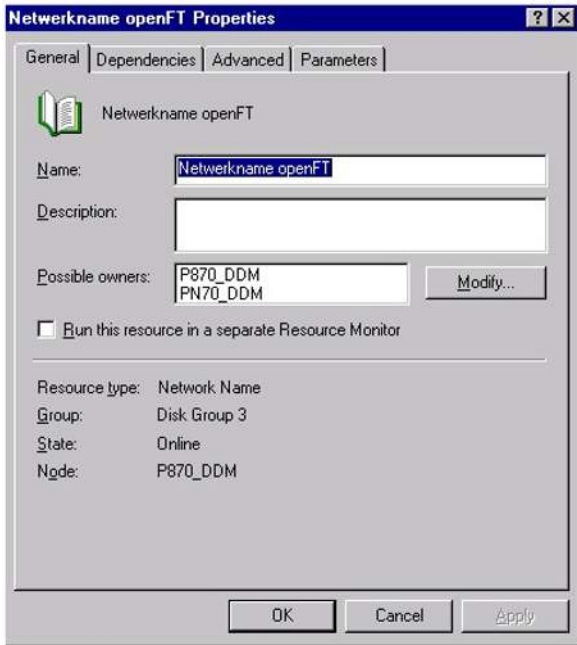
Configure openFT in a cluster: IP Address - Advanced



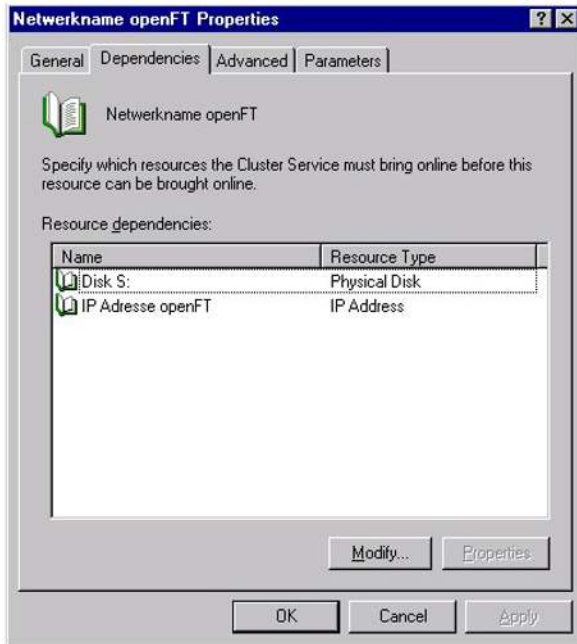
Configure openFT in a cluster: IP Address - Parameters

2.	Name	e.g. Network name openFT
	Resource Type	Network Name
	Dependencies	Physical Disk (in this case Disk S:) IP address openFT
	Advanced	use standard or customize
	Parameters	Name e.g. OPENFT

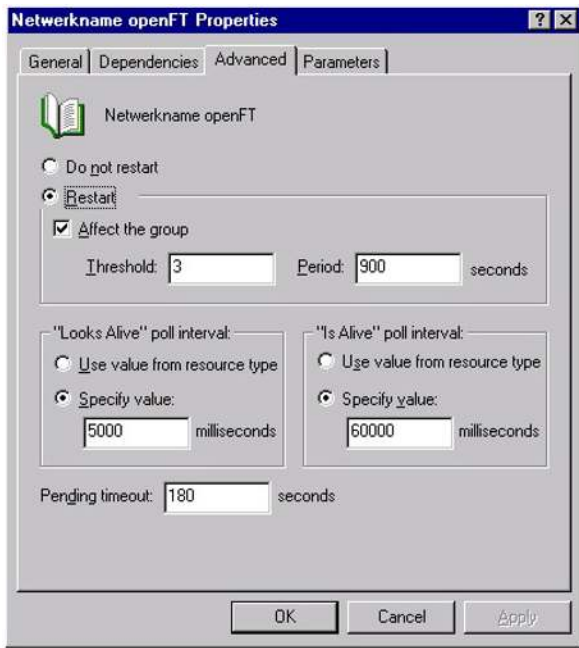
Press *Apply* to create the resource and bring it online (right mouse button - *Bring Online*).



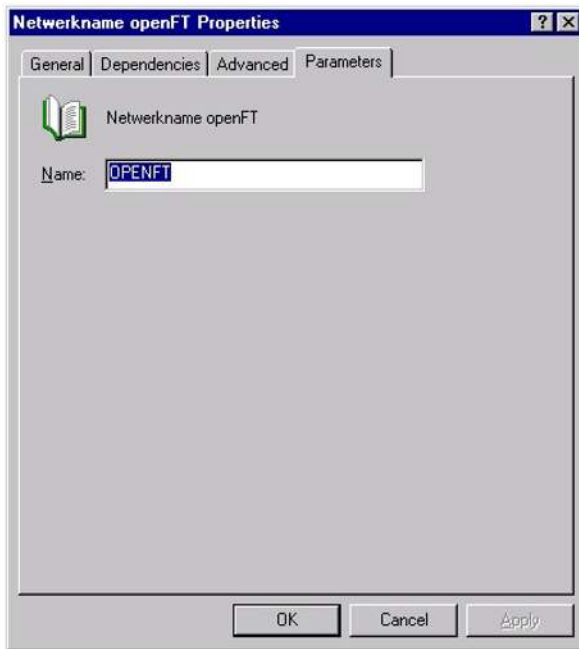
Configure openFT in a cluster: Network Name - General



Configure openFT in a cluster: Network Name - Dependencies



Configure openFT in a cluster: Network Name - Advanced



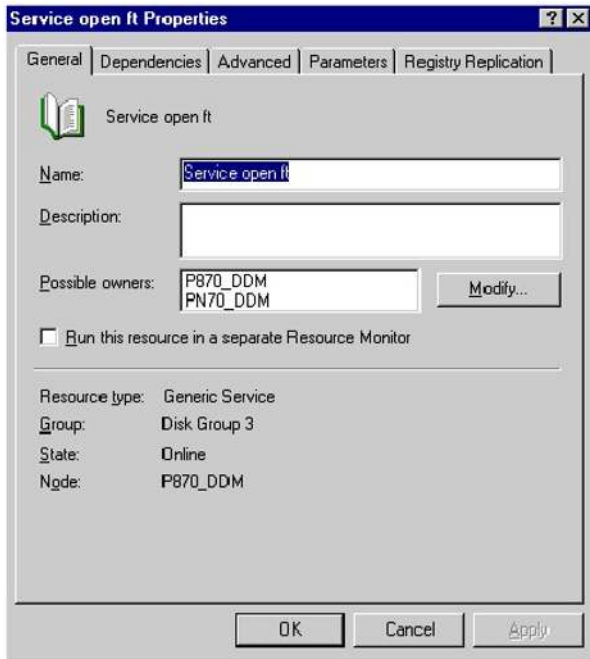
Configure openFT in a cluster: Network Name - Parameters

Configuration openFT service:

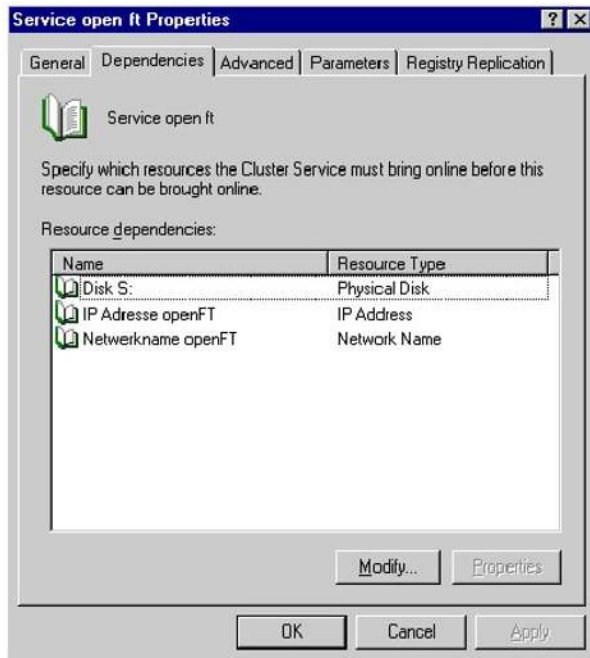
3.	Name	e.g. Service openFT
	Resource Type	Generic Service
	Dependencies	Physical Disk (in this case Disk S:) IP address openFT Network name openFT

Advanced	use standard or customize
Parameters	Service name: openFT - cluster
Registry Replication	SOFTWARE\Classes\SNI\WINTNS (only necessary if TNS is used)

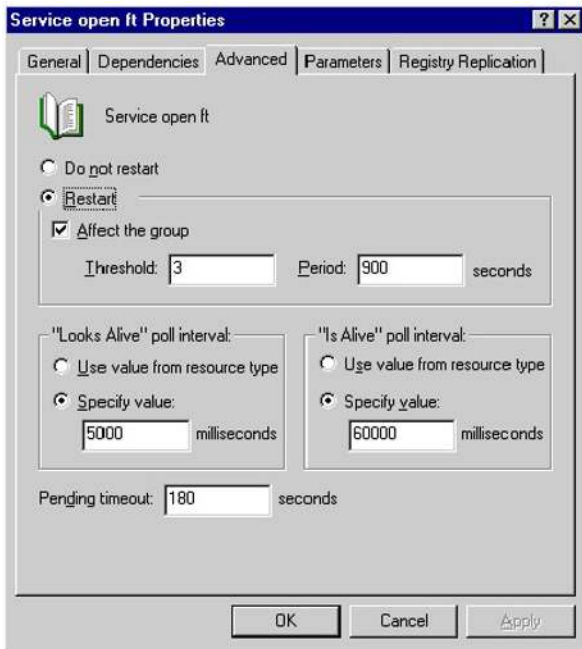
Press *Apply* to create the resource and bring it online (right mouse button – *Bring Online*).



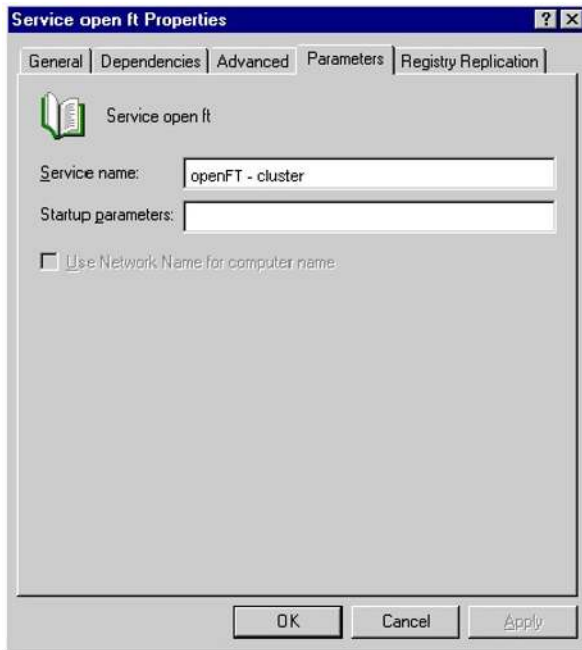
Configure openFT in a cluster: Generic Service - General



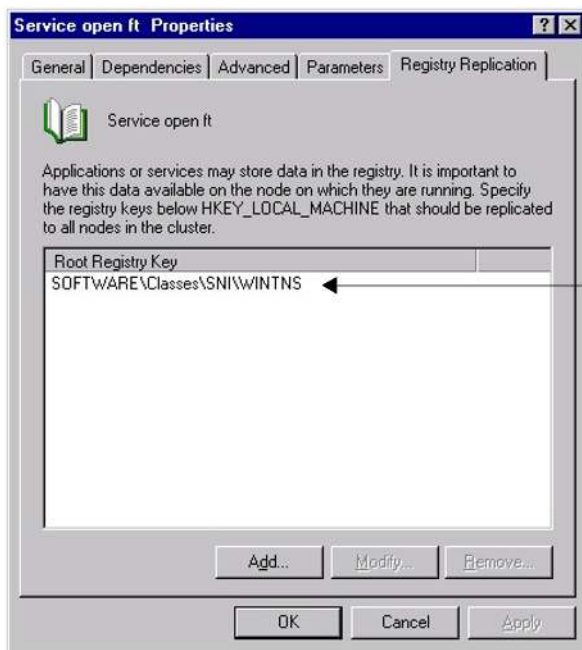
Configure openFT in a cluster: Generic Service - Dependencies



Configure openFT in a cluster: Generic Service - Advanced



Configure openFT in a cluster: Generic Service - Parameters



Only necessary if CMX and TNS are used

Configure openFT in a cluster: Generic Service - Registry Replication

If CMX and TNS are not used, Registry Replication is not necessary.

7.5 FarSync X.25 transport system under Linux and Windows systems

FarSync X.25 cards from the manufacturer FarSite are directly supported by openFT under Linux and Windows, i.e. the configuration data is specified completely in openFT, PCMX is no longer required for this. CMX operation must be switched off (*ftmodo -cmx=n*).

The coupling method XOT (X.25 via TCP/IP) is also supported under Linux by using the FarSync XOT Runtime.

The X.25 transport system can be used by the openFT and FTAM protocol.

7.5.1 Direct support of the FarSync X.25 for Windows systems

Under Windows systems a maximum of 4 FarSync X.25 cards (adapters) with a maximum of 4 lines each are supported. All in all, 16 lines are supported. Adapters and lines are numbered from 0 to 3. Each line is thus uniquely identified by a combination of adapter number and line number (0-3).

The respective latest version of FarSync X.25 software for Windows systems is recommended for use with openFT. This is currently:

- 4.2.0 as of Windows 7

7.5.2 Support of the FarSync X.25 for Linux systems

Under Linux a separate device is configured for each line. A maximum of 16 lines are supported under Linux. The lines are numbered from 0 to 15. Each line is thus uniquely identified via its own number.

The minimum required version of FarSync X.25 software for Linux is 3.2.9. The required X.25 software can be downloaded from the following site:

http://www.farsite.com/custsupp/Download_X.25_software.htm.

XOT support

At least FarSync XOT Runtime version 3.2 is required for the support of XOT (X.25 via TCP/IP) under Linux. A coupling to the FarSync X.25 Gateway or another device from any manufacturer which does support XOT is thus possible.

The FarSync XOT Runtime provides as default under Linux a virtual X.25 interface. The configuration of openFT for using FarSync XOT Runtime is identical with the configuration for using FarSync X.25 cards. In order to address XOT Runtime only the adapter number (standard value: Adapter 0) of the XOT adapter (of the virtual X.25 interface) must be specified. An entry for XOT routing must also be created in the FarSync configuration program, i.e. for outgoing connections, an allocation must be created between a DTE target address and the IP target address of the XOT partner.

i Manual changes in the configuration files of the XOT-Runtime enable the use of several XOT adapters. Due to problems in the XOT-Runtime it is not recommended to use multiple XOT adapters for openFT.

7.5.3 Configuring the FarSync X.25 transport system in openFT

You configure the FarSync X.25 transport system using the *ftmodo* command.

A separate DTE address can be assigned to each X.25 line on the FarSync X.25 card, which is sent with outgoing connections as DTE address of the sender (Calling DTE Address) in the X.25 connection set-up. The assignment of a DTE address to a line is optional. If no DTE address can be assigned to a line, then no DTE sender address is sent in the X.25 connection set-up.

7.6 Sample files

openFT is supplied with a range of sample files that you can use for various purposes. Once openFT has been installed, you will find these files in the directory

- */opt/openFT/samples* (Unix systems)
- *openFT-installation-directory\samples* (Windows systems)

ftadm

The file *config.xml* contains a simple sample configuration for remote administration. You can use this sample as a template and adapt it according to your needs.

ftscript

This directory contains examples for the openFT-Script interface. You will find a description of the interface in the manual "openFT (Unix and Windows systems) - openFT-Script Interface".

filedist.ftsc

Distribute files to several different partner systems.

transsuc.ftsc

Transfer a file to a partner system with follow-up processing.

treecopy.ftsc

Transfer a complete directory tree to a partner system.

ftacct.xlt (Unix systems)

The Excel template demonstrates how to evaluate the CSV output format of the logging commands and how to use them in Excel for accounting purposes.

ftapi

Comprises the files *sample1.c*, *sample2.c*, *sample3.c*, *sample4.c*, *sample5.c*, *sample6.c* and *sample7.c*.

These examples illustrate various options for using the C programming interface of openFT. You will find a description of the examples in the manual "openFT (Unix and Windows systems) - Program Interface".

sample1.c

Transfer a file asynchronously

sample2.c

Transfer several files with follow-up processing.

sample3.c

Show the contents of a remote directory.

sample4.c

Execute a command on the partner system.

sample5.c

Run a loop that reads in, in quantities equivalent to the size of the buffer, the file attributes of all the files in a remote directory.

sample6.c

Create a directory on partner system.

sample7.c

Delete an empty directory on partner system.

java

Comprises the files *Sample1.java*, *Sample2.java*, *Sample3.java*, *Sample4.java*, *Sample5.java*, *Sample6.java* and *Sample7.java*.

These examples illustrate the Java programming interface of openFT. How to compile and run the examples is described in the manual "openFT (Unix and Windows systems) - Program Interface".

Sample1.java

Transfer a file asynchronously

Sample2.java

Transfer several files with follow-up processing.

Sample3.java

Show the contents of a remote directory.

Sample4.java

Execute a command on the partner system.

Sample5.java

Run a loop that reads in, in quantities equivalent to the size of the buffer, the file attributes of all the files in a remote directory.

Sample6.java

Create a new directory on partner system.

Sample7.java

Delete an empty directory on partner system.

treecopy-get, treecopy-send, treecopy-send-unique (Unix systems)

These shell scripts illustrate various ways of transferring a complete directory to Unix or Windows partner systems.

treecopy-get

Fetch all files of a directory from a partner system using preprocessing. In this example, preprocessing is used in the remote system without an intermediate file being specified.

treecopy-send

Pack all files of a directory in a tar archive using preprocessing, transfer them to a partner system and unpack them there using postprocessing.

treecopy-send-unique

Pack all files of a directory in a tar archive using preprocessing, transfer them to a partner system and unpack them there using follow-up processing.

The use of %UNIQUE in the receive file name allows several scripts to be executed concurrently.

msexcel (Windows systems)

This directory contains the Microsoft Excel file *ftaccnt.xlt* and *openft32.xls*.

ftaccnt.xlt

Microsoft Excel template (Microsoft Excel 2003 and 2007). The template demonstrates how to evaluate the CSV output format of the logging commands and how to use them in Microsoft Excel for accounting purposes.

openft32.xls

Adds the menu *openFT* containing the commands *Transfer this File* and *Cancel Transmission* to the menu bar of Microsoft Excel. The file contains the associated Microsoft Visual Basic macros. The description of the macros is displayed when you open the file.

msword (Windows systems)

The file *openFT32.dot* is a Microsoft Word template (Microsoft Word 2003 and 2007). Documents created with this template can transfer themselves to a partner system where they can be printed if necessary. To do this, the template must be installed in the folder for Microsoft Word templates locally on your computer and on the partner system.

openFT32.dot adds the menu *openFT* with the commands *Transfer Document*, *Cancel Transfer* and *Send Clipboard* to the menu bar of Microsoft Word. The file contains the associated Microsoft Visual Basic macros. The description of the macros is displayed when you open the file.

ocxdemo (Windows systems)

This sample illustrates the use of the OCX control *fttrans.ocx*. The directory contains the relevant Microsoft Visual Basic application and the associated source code. You will find a description of OCX control in openFT in the Readme file in the ocxdemo directory and in the manual "openFT (Unix and Windows systems) - Program Interface".

The Microsoft Visual Basic application is started and the OCX control for openFT is loaded by calling *ocxdemo.exe*.

shellext (Windows systems)

This directory contains the program library and the install file and uninstall file for the openFT shell extension. The openFT shell extension allows you to create predefined send patterns on the desktop in order to start a file transfer to a defined partner by dragging and dropping the file onto the send pattern from the Windows Explorer.

After installation, you create a new send pattern as follows:

-
- Choose *New* from the desktop context menu and then choose the entry *openFT Send Pattern*.
A new icon whose name you can change is created on the desktop. The file name extension *.openFTst* must not be deleted from the name. You can then configure the openFT-specific parameters.
 - Right-click the send pattern icon and choose *Properties* from the menu. The *Properties* dialog box is displayed. This contains the following additional openFT-specific tabs:

openFT General

Specifications on the partner system

openFT Options

Specifications on the send request options

simple (Windows systems)

The file *ncopy.c* illustrates how to call the commands from a program using the associated DLLs using the *ncopy* command as an example.

www (Windows systems)

The sample programs in this directory show how you can use openFT for downloading in the Internet or in an intranet. The example uses a Windows system as a client and a Unix system as the server platform. You will find a detailed description of the concept and the way in which it is implemented in the *Readme* file in the *www* directory.