English

FUJITSU

openFT V12.0 for Unix Systems

Installation and Administration

System Administrator Guide

Edition September 2012

Comments… Suggestions… Corrections…

The User Documentation Department would like to know your opinion on this manual. Your feedback helps us to optimize our documentation to suit your individual needs.

Feel free to send us your comments by e-mail to:
manuals@ts.fujitsu.com

Certified documentation according to DIN EN ISO 9001:2008

To ensure a consistently high quality standard and user-friendliness, this documentation was created to meet the regulations of a quality management system which complies with the requirements of the standard DIN EN ISO 9001:2008.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright and Trademarks

Contents

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

- BS2000/OSD®
- Solaris™ (SPARC®/Intel™), LINUX®, AIX®, HP-UX®
- Microsoft® Windows Vista™, Windows™ 7, Windows Server 2008™ and Windows Server 2008 R2™
- z/OS (IBM®)

1.1 Brief description of the product

openFT for Unix systems is the file transfer product for systems with a Unix based operating system.

All openFT products communicate with each other using the openFT protocol (previously known as the: 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.

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

When used in combination with openFT-FTP, openFT also supports the FTP protocol. This makes it possible to interconnect with other FTP servers.

With the integrated FTAC function, openFT offers extended admission and access protection (FTAC stands for **F**ile **T**ransfer **A**ccess **C**ontrol).

1.2 Target group

This manual contains the information which is needed by openFT and FTAC administrators of Unix systems for their work and which is not included in the User Guide.

For general information on file transfer and file management, you will also need the User Guide. Further literature is listed in the references.

The manual covers Oracle Solaris systems as well as portings to other Unix platforms. The operating system-dependent differences are described in detail in the Release Notices supplied on the respective product CD.

1.3 Concept of openFT for Unix systems manuals

The complete description of openFT and its optional components comprises four manuals. The description is divided among the manuals as follows:

- openFT for Unix systems - Installation and Administration

  The system administrator manual is intended for FT, FTAC and ADM administrators. It describes:
  - the installation of openFT and its optional components
  - the operation, control and monitoring of the FT system and the FTAC environment
  - the administration commands for FT and FTAC administrators
  - the configuration and operation of a remote administration server and a ADM trap server
  - important CMX commands.

- openFT for Unix systems - Managed File Transfer in the Open World

  The user manual is intended for the openFT user and describes:
  - the basic functions of the openFT product family,
  - the conventions for file transfers to computers running different operating systems,
  - details on implementing FTAM,
  - the openFT user commands,
  - the openFT-Script commands,
  - the messages of the different components.

- openFT for Unix systems and Windows systems - C Program Interface

  This manual is intended for C programmers and describes the C program interface on Unix systems and Windows systems.

- openFT for Unix systems and Windows systems - openFT-Script Interface

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

Many of the functions described in the manuals are also available in the openFT graphical interface, the openFT Explorer. 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 of the manual

This section describes the changes in openFT V12.0 for Unix systems compared to openFT V12.0 for Unix systems.

The functional extensions to the openFT commands, whether they relate to administrators or users, are also available in the openFT Explorer. For details, see the *New functions* section in the associated online help system.

Configuration Editor for remote administration

With the new Configuration Editor, openFT provides a graphical user interface which can be used to create or modify a configuration file for remote administration. The configuration can be seen immediately in the Configuration Editor in the form of a tree structure and corresponds to the subsequent display in the openFT Explorer.

The Configuration Editor is started via the openFT Explorer.

Extended logging functions

The logging functions have been extended as follows:

- Switch log file and offline logging

  The log file can be changed during operation. After switchover, new log records are written to a new log file. The previous log file is retained as an offline log file. The log records it contains can still be viewed using the tools available in openFT.

  To permit this, the command interface has been extended as follows:

  – *ftmodo*:

    New option *-lf=c* to switch the log file.

  – *ftshwl*:

    New options *-lf*, *-tlf* and *-plf* to view the log records present in offline log files.

    New option *-llf* to output the names of all log files (including offline log files).

  – *ftdell*:

    New selection criterion *-tlf* to delete offline log records.

- Automatic deletion of log records

  Intervals for the automatic deletion of log records can be set in the operating parameters. To make this possible, the following options have been added to the *ftmodo* command: *-ld*, *-lda*, *-ldd* and *-ldt*. The settings can be displayed using the *ftshwo* command.

- Polling function for the output of log records

  The new options *-po* and *-pnr* in the *ftshwl* command can be used to set the interval and number of repetitions (polling).

- Wildcards for partner names during the output of log records

  In the *ftshwl* command, it is also possible to use the wildcards "*" and "?" when specifying the partner name (*-pn=*).

Enhanced security functions

- Import keys

  The new command *ftimpk* can be used to import both externally generated private keys and the public keys of partner systems.

- Expiration data and authentication level of RSA keys

  – Using the new command *ftmodk*, it is possible to define an expiration date and modify the authentication level (1 or 2) for keys that are used for the authentication of partner systems.

  Authentication level 2 was introduced with openFT V11.0B and meets higher security requirements.

  – The new command *ftshwk* can be used to output the attributes of the keys stored in the system.

  – *ftshwl* displays the authentication level (output parameter SEC-OPTS, new values LAUTH2 and RAUTH2).

- Force data encryption

  The new option *-c* in the *ftmodo* command can be used to force data encryption for file transfer and administration requests. The settings can be made separately for inbound and outbound requests.

- Following installation, openFT uses an RSA key of length 2048 bits by default.

- PAM support

  The Pluggable Authentication Modules (PAM) as authentication services for password encryption in openFT are supported for all platforms. Support for Solaris was already present in V11.0 but was not yet described in the manual.

- File access and admission check under user permissions

  All accesses and admission checks by openFT relating to a user's files and directories are performed under the permissions of the relevant user.

Extended partner management

- Partners in the partner list can also be explicitly deactivated for inbound requests.

  This is possible using the new option *-ist* in the *ftaddptn* and *ftmodptn* commands. In *ftshwptn*, the current state (activated/deactivated) is displayed in the output parameter INBND.

- Serialization of asynchronous outbound requests to specific partners

  The new option *-rqp* in the *ftaddptn* and *ftmodptn* commands makes it possible to control whether asynchronous outbound requests to a specific partner should always be run serially or whether parallel connections are also permitted. In the *ftshwptn* command, this attribute is displayed in the output parameter REQU-P.

Extended request management

- Global request ID

  In the event of an FT request, the initiator's request number is transferred to the responder where it is visible as a global request ID. This means that any request can be unambiguously assigned to an initiator and responder.

  The *ftshwr* and *ftshwl* commands have been extended as follows:

  – At the responder, the global request ID is displayed in the new output parameter GLOB-ID in each command.

  – The new parameter *-gid*, makes it possible to perform selection on the basis of a global request ID in both commands.

Operation with and without CMX

The new option *-cmx* in the *ftmodo* command can be used to switch between the operating modes "with CMX" and "without CMX". The current mode is displayed in the output parameter USE CMX of the *ftshwo* command.

Following installation, the operating mode "without CMX" is set.

Extended diagnostics

The new option *-troll* in the *ftmodo* command can be used to activate and deactivate the trace for the lower protocol layers and control the scope of the trace during operation.

The current setting is displayed in the output parameter OPTIONS-LL (line FUNC) in the *ftshwo* command.

Extension to the C programming interface and the openFT-Script interface

The programming interface has been extended by the following function groups:

- *ft_sd** to determine the attributes of all the files in a directory in the remote system.
- *ft_xc** for the synchronous execution of a command in the remote system.

The openFT-Script interface has been extended by the following commands for the variable storage of openFT-Script requests:

- *ft_modsuo* for modifying openFT-Script user options.
- *ft_shwsuo* for displaying openFT-Script user options.

Integration in Solaris SMF

On Solaris systems, openFT is integrated in the Service Management Facility (SMF) concept:

- Both installation and the *ftstart*, *ftstop*, *ftcrei* and *ftdeli* commands have been adapted to the SMF procedure.
- The *ftalarm* manifest is now also installed for each instance.

Other changes

- The *ft* and *ncopy* commands now have the additional alias names *ftacopy* (for *ft*) and *ftscopy* (for *ncopy*) in order to avoid confusion with operating system commands or commands used by other vendors.
- The *ftinfo* command has been extended and now outputs additional information.
- The maximum record length on file transfer requests and when setting local file attributes has been extended to 65535. This affects the following commands and options:
  - *ncopy -r=*
  - *ft -r=*
  - *ftmodf -rl=*
- On Solaris systems, openFT permits installation in an alternative root directory.
- Migration assistance for elimination of TNS

  The tool *tns2ptn* is available for users who want to switch to operation without TNS. *tns2ptn* is used to generate commands which can be used to create appropriate entries in the partner list on the basis of TNS entries with the RFC1006 address format.

- The description of dynamic partners is now more precise. To this end, the partner types "named partner", "registered dynamic partner" and 'free dynamic partner" have been introduced.
- The description of the CSV output for the SHOW commands (*ftshw*, *ftshwa*, etc.) has been greatly extended.

Obsolete functions

- The BSFT interface is no longer supported. The associated section in the manual "openFT for Unix Systems - Managed File Transfer in the Open World" has been removed.

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.


: indicates notes


: Indicates warnings.

Additional conventions are used for the command descriptions, see page 162.

1.6 README files

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

1.7 Current information on the Internet

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

1.8 License provisions

The following provisions apply to the use of *libxml2* and Secure FTP and xerces-J for openFT-Script.

Use of libxml2

libxml2 is used for processing XML data. This contains the XML C Parser and an XML toolkit. *libxml2* was originally developed for the Gnome project, but can also be used outside Gnome. *libxml2* is freeware available under the MIT license:

```
Copyright (c) <2008> <Daniel Veillard>

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

Use of openSSL for Secure FTP

The following provisions apply to the use of Secure FTP.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/). This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

```
LICENSE ISSUES
==============

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the
OpenSSL License and the original SSLeay license apply to the toolkit. See below
for the actual license texts. Actually both licenses are BSD-style Open Source
licenses. In case of any license issues related to OpenSSL please contact
openssl-core@openssl.org.

OpenSSL License
---------------

================================================================================
```

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscapes SSL. This library is free for commercial and non-commercial use as long as the following conditions are aheared to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
   "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)"
   The word 'cryptographic' can be left out if the rouines from the library being used are not cryptographic related :-).
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement:
   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

Use of xerces-J for openFT-Script

The following provisions apply to operation with openFT-Script.

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

```
/* ====================================================================
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 2000 The Apache Software Foundation.  All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. The end-user documentation included with the redistribution,
 *    if any, must include the following acknowledgment:
 *       "This product includes software developed by the
 *        Apache Software Foundation (http://www.apache.org/)."
 *    Alternately, this acknowledgment may appear in the software itself,
 *    if and wherever such third-party acknowledgments normally appear.
 *
 * 4. The names "Apache" and "Apache Software Foundation" must
 *    not be used to endorse or promote products derived from this
 *    software without prior written permission. For written
 *    permission, please contact apache@apache.org.
 *
 * 5. Products derived from this software may not be called "Apache",
 *    nor may "Apache" appear in their name, without prior written
 *    permission of the Apache Software Foundation.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED.  IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
```

```
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
* ====================================================================
*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation.  For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.
*
* Portions of this software are based upon public domain software
* originally written at the National Center for Supercomputing Applications,
* University of Illinois, Urbana-Champaign.
*/
```

2 Installation

This chapter describes the installation and configuration of openFT.

openFT is shipped with a communications manager. 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).

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 < V10.0 on it.
- Update installation
  Your computer has openFT version V10.0 or V11.0 installed.
- Installation of a correction version
  Your computer has openFT version 12.0 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 language interface” on page 61.

- If you want to encrypt file contents, you must also install openFT-CR V12.0 for Unix systems. This software is offered without a license at a fixed price. If an openFT-CR version < V10.0 is already installed, then you must first uninstall this version before installing openFT. You may only install openFT-CR V12.0 after openFT V12.0 has been installed.
- If you want to use the openFT-Script interface or the Java API then the J2SE™ Runtime Environment 5.0 (JRE 5.0) 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 above-mentioned 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 default instance named *std* always exists.

  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 < V10.0 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*.
   - Uninstall openFT-CR, openFT and the add-on products.
2. Install the openFT V12 product software.

   When doing this, please note the following:

   On a system in which the openFT installation takes place in a dialog, you need to answer a question during installation asking you if you have a valid openFT-FTAM license and/or a valid openFT-FTP license. Only activate this option if you have a valid license for openFT-FTAM or openFT-FTP! Depending on the responses, openFT-FTAM and/or openFT-FTP is installed or not.

   This question is 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 (see also section “install.ftam - Install openFT-FTAM” on page 366 and section “install.ftp - Install openFT-FTP” on page 367).

3. Import the saved admission sets and admission profiles using *ftimpe*. If the admission sets and admission profiles were exported from an openFT version < V8.1then 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.

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 the section “TNS entries created automatically” on page 420).

  If CMX is subsequently installed then you can also use a tool to create default TNS entries at some later date, see page 419.

- The instance directory for the default instance is set up, see page 26.

  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” on page 38.

  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 page 80).
- 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” on page 96).

  On the Solaris platform, SMF is supported as of openFT V12.0, see section “Solaris SMF” on page 41. 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 on page 26.

2.1.2 Update installation from openFT V10.0 and V11.0

If openFT V10.0 or V11.0 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 default 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 page 364).

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 (see also section “install.ftam - Install openFT-FTAM” on page 366 and section “install.ftp - Install openFT-FTP” on page 367).

3. 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” on page 41.

   See also section “openFT instances and cluster operation” on page 96.

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/etcinit/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 “Solaris SMF” on page 41. 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 (in the case of an update from V11.0).

  - In the case of an update installation from V10.0 to V12.0, the FTP server is activated if a port a number other than 0 was set for the FTP server previously.

- 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 on page 26.

2.1.3 Installation of a patch

Installation of a patch means that openFT V12.0 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 V12.0 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.

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.
- The FT 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 V12 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 V10.0 or V11.0, or, if a correction version is to be installed, an openFT Version V12.0.

If openFT < V10.0 is installed on a system then installation must not be performed in an alternative root directory.

Proceed as follows:

1. Install the openFT V12 product software in the alternative root directory. The installation method is described in the Release Notice.

   When you do this, the fixed files and directories in the openFT package are installed in the alternative root directory, e.g. */altroot/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 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 V12 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 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

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'
```

2.2 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 page 38
- 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 page 39
- disabling automatic startup of openFT, see page 40
- activating *ftalarm* function, see page 40
- starting openFT subagents automatically, see page 41
- installing or uninstalling openFT-FTAM under HP-UX, AIX and Linux, see page 45
- installing or uninstalling openFT-FTP under HP-UX, AIX and Linux, see page 45
- activating/disabling authentication via PAM (Pluggable Authentication Modules), see page 46
- creating the partner list from TNS, see page 48
- configuring the remote administration server
  If you want to use your system as a remote administration server, you must configure the server. See the section “Configuring the remote administration server” on page 117.
- configuring the ADM trap server
  If you want to use your system as an ADM trap server, you must configure the server. See the section “Configuring the ADM trap server” on page 147.
- creating TNS entries
  If you use the TNS you may need to create the TNS entries, see the section “Entering transport system applications in the TNS” on page 419.

  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 “Creating default TNS entries via a script” on page 419.

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

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” on page 91).

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 KEY-LEN CCS-NAME
  YES     NONE       16        8      2000       30       65535   2048   ISO88591
PTN-CHK DYN-PART SEC-LEV  FTAC-LOG FT-LOG ADM-LOG     USE TNS USE CMX ENC-MAND
  STD        ON   B-P-ATTR    ALL    ALL    ALL          NO      NO        NO
OPENFT-APPL      FTAM-APPL          FTP-PORT            ADM-PORT         ADM-CS
*STD             *STD               21                  11000              NO
ACTIVE           ACTIVE             DISABLED            ACTIVE
HOST-NAME        IDENTIFICATION / LOCAL SYSTEM NAME
*NONE            mc011.mynet.local / $FJAM,MC011

DEL-LOG  ON  AT     RETPD    ADM-TRAP-SERVER
 OFF   DAILY 00:00    14     *NONE

TRAP: SS-STATE FT-STATE PART-STATE PART-UNREA RQ-STATE TRANS-SUCC TRANS-FAIL
CONS    OFF      OFF       OFF        OFF       OFF       OFF        OFF
ADM              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 page 331.

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 “Instance Identifications” on page 79.

- 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 -dn=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* 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 page 221. 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

This section only applies to Solaris, because openFT V12 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” on page 41.

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” on page 96).

These files then start the corresponding 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). 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” on page 96).

These files then terminate the corresponding openFT instance.

2.2.4 Enabling the ftalarm command

This section does not apply to Solaris where a different mechanism is used, see section “Activating ftalarm automatically” on page 42.

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 page 180. 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 Starting the openFT subagent automatically

This section does not apply to Solaris where another mechanism is used, see section “Solaris SMF” on page 41.

If you want to automatically start the openFT subagent for administration using SNMP at system startup, you must activate the corresponding line with the *ftagt* command in the startup and shutdown file */var/openFT/std/etcinit/openFTinst* and/or in the startup and shutdown files of other instances.

More details on this can be found in the chapter “Administering openFT via SNMP” on page 103.

Please note for clusters that SNMP can only work with a single openFT instance. The deciding factor is which instance is set up when the agent is started (see also section “openFT instances and cluster operation” on page 96).

2.2.6 Solaris SMF

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 section “Solaris SMF” on page 42.
- 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, 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.

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.

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
offline        16:32:14 svc:/application/ftalarm:inst001
#. ftseti inst001
# ftstart
ftstart: openFT 12.0A00 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
   ```

Monitoring the openFT instance via SNMP

With *ftagt*, you can monitor precisely one openFT instance via SNMP. On Solaris systems, *ftagt* is administered via SMF.

You can

- view the current SNMP instance that is to be monitored:

```
# svcprop -t -p method_context/environment ftagt
method_context/environment astring OPENFTINSTANCE=std
```

- display the status of *ftagt*:

```
# svcs ftagt
STATE          STIME    FMRI
disabled       Jul_11   svc:/application/ftagt:default
```

- enable *ftagt* for the instance set using OPENFTINSTANCE:

```
svcadm enable ftagt
```

- terminate *ftagt* for the instance set using OPENFTINSTANCE:

```
svcadm disable ftagt
```

- change the instance that is to be monitored (e.g. to *hugo*):
  - disable *ftagt*

    ```
    svcadm disable ftagt
    ```

  - Modify the environment

    ```
    svccfg -s ftagt:default setenv -i OPENFTINSTANCE hugo
    ```

  - Activate the environment

    ```
    svcadm refresh ftagt
    ```

  - Enable *ftagt* for *hugo*

    ```
    svcadm enable ftagt
    ```

An openFT instance can only be administered with SNMP if

- the openFT instance exists
- the openFT instance has been started
- OPENFTINSTANCE is correctly set for *ftagt*
- *ftagt* is started

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

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 section “install.ftam - Install openFT-FTAM” on page 366.

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.

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

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 page 367.

Installation is only permitted if a valid openFT-FTAM 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.

2.2.9 Authentication via PAM

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 "Enabling/disabling the PAM function".

In many cases, it is necessary to check the configuration files and adapt the entries, see "Checking and modifying the PAM configuration files".

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
    ```

2.2.10 Creating the partner list from TNS

Thanks 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:

   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 Tasks of the administrator

This chapter describes the most important administration tasks to be performed when running openFT. You can administer openFT both via the openFT Explorer and by using commands. The following options are available:

– Functions and commands that only the administrator may use (e.g. start openFT or delete log records),

– Functions and commands that are accessible to both the user and the administrator, but where the administrator is allowed to do more than the user (e.g. modify admission sets).

The tasks of the administrator include:

– Setting operating parameters[1) 2)]
– Starting and stopping openFT[1) 2)]
– Administering the request queue[1)]
– Viewing and deleting log records[1)]
– Administering admission sets and FT profiles[1)]
– Diagnostic options, e.g. switching the trace for error diagnostics on and off[1) 2)]
– Creating and administering instances in order to use openFT in the cluster
– Creating key pair sets [1)] and making a current public key available to the partner systems. This enables the local system to be authenticated by the partner.
– Obtaining the public keys of partner systems and suitably storing them in the local system so that the partner systems can be authenticated by the local system.

The administration functions marked with [1)] can also be executed via the openFT Explorer), provided an X terminal or corresponding emulation is available. More information on the openFT Explorer can be found in the manual on “openFT V8.1 for Unix systems” and in the online help.

The administration functions marked with [2)] can also be performed via an SNMP management station, see chapter “Administering openFT via SNMP” on page 103.

The administration of the FTAC functions can also be transferred to another person, known as the FTAC administrator. Central administration including setting up a remote administration server is a separate task. See page 51 and the chapter “Central administration” on page 111.

Who is the FT administrator?

openFT can be administered by all user IDs that have root permission (UID=0), i.e. all user IDs with UID=0 have FT administrator permission.

Who is the FTAC administrator?

Following a new installation, the openFT and FTAC administrators are identical. This means that all users who possess FT administration rights on the system are also FTAC administrators. The FTAC administrator is identified by the fact that the corresponding privilege is defined in his or her admission set. You can transfer this property to another login name by using the *ftmoda* command. This is useful, for example, if someone other than the system administrator is responsible for data security. The FTAC administrator has the following permissions:

- administer admission profiles, see page 93
- administer admission sets, see page 91
- back up the FTAC environment, see page 95

In addition, the FTAC administrator can also administer logging as well as the FT administrator and the ADM administrator, see page 88.

Depending on the user ID under which it is set up, the FTAC administrator account has various rights and options:

- Default setting (FT administrator is FTAC administrator)
  Every other user ID that possesses FT administrator permissions is also an FTAC administrator. This means that every FTAC administrator has the permissions of an FT administrator.
- Transfer of the FTAC privilege to a different user ID with FT administrator permissions:
  This means that only this user ID still has both FT and FTAC administrator permissions. All other previous FT administrators lose their explicit FTAC administrator permissions.
- Transfer to a user ID without FT administrator permissions:
  An FT administrator is no longer permitted to administer any admission sets and admission profiles or to back up the FTAC environment. The FTAC administrator only has the FTAC administrator privileges listed above, but not the permissions of an FT administrator.

The command *ftmoda @ftadm -priv=y* allows both FTAC administrators and FT administrators to reset FTAC administration to the default settings, i.e. FT administrators and FTAC administrators are identical again.

ADM administrator

The ADM administrator is the only person permitted to administer the remote administration server. Working with a remote administration server and the role of the ADM administrator are described in detail in the chapter “Central administration” on page 111. Immediately after a new installation, no ADM administrator yet exists. The FTAC administrator must first define one. See the section “Defining the ADM administrator” on page 118.

3.1 Setting 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 instance identification of the local openFT instance.
- 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 scope for protocols during openFT operation.
- The length of the RSA key to be used for encryption purposes.
- The code table that should be used by default for local text files.

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 *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:

On 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.

3.2 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

Name of the CCS	Meaning
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 an blank. The remainder of the line contains a comment which characterizes the code contained.

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

- The third line contains three numbers.

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

  The second number is currently always '0'.

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

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

  – The first of these lines contains the number of the code page in the form of a two-digit hexadecimal number.

  – 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
000000010002000300040005000600070008000900A000B000C000D000E000F
001000110012001300140015001600170018001900 1A001B001C001D001E001F
002000210022002300240025002600270028002900 2A002B002C002D002E002F
003000310032003300340035003600370038003900 3A003B003C003D003E003F
004000410042004300440045004600470048004900 4A004B004C004D004E004F
005000510052005300540055005600570058005900 5A005B005C005D005E005F
006000610062006300640065006600670068006900 6A006B006C006D006E006F
007000710072007300740075007600770078007900 7A007B007C007D007E007F
008000810082008300840085008600870088008900 8A008B008C008D008E008F
009000910092009300940095009600970098009900 9A009B009C009D009E009F
00A000A100A200A320AC00A5016000A7016100A900AA00AB00AC00AD00AE00AF
00B000B100B200B3017D00B500B600B7017E00B900BA00BB0152015301780 0BF
00C000C100C200C300C400C500C600C700C800C900CA00CB00CC00CD00CE00CF
00D000D100D200D300D400D500D600D700D800D900DA00DB00DC00DD00DE00DF
00E000E100E200E300E400E500E600E700E800E900EA00EB00EC00ED00EE00EF
00F000F100F200F300F400F500F600F700F800F900FA00FB00FC00FD00FE00FF
```

3.3 Starting and stopping openFT

By default, openFT (i.e. the asynchronous openFT server) is started automatically at system startup.

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 the section “Disabling the automatic startup of openFT” on page 40).

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

If the asynchronous openFT server is not started, only synchronous requests are executed. Asynchronous requests are stored in the request queue. Furthermore, no further requests are accepted from partner systems.

After being started, the asynchronous openFT server executes both asynchronously issued requests as well as file transfer requests issued on the remote system.

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

3.4 Setting the protection bit for newly created files

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 *svcprop* command (here for the default instance):

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

   Output:

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

3.5 File access under user rights

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.

3.6 Switching the language interface

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, see page 220. 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 LANG
Invalid value or a language that is not installed	Not evaluated	Default setting
Valid value (de or en)	Not evaluated	Language set in OPENFTLANG

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.

3.7 Administering requests

The request queue stores all asynchronous outbound requests, and all inbound requests. As the administrator, you can

- **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. You can display the request queue with the *ftshwr* command.
- **modify** the **processing order** of all requests on your system, including those of other users. You can do this by using the *ftmodr* command.
- **cancel** asynchronous requests on your system, including those of other users. You can do this by using the *ftcanr* command.

You can also view the request queue in the openFT Explorer by clicking on the *Request Queue* object directory. In addition, you can also execute the following functions via the openFT Explorer:
– Cancellation of asynchronous requests
– Update the request queue
– Change the priority of requests
– 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.

3.8 Administering partners

openFT allows you to perform file transfers with a number of different partner systems. These partner systems may be accessible via different transport systems and protocols. To allow you to administer these partner systems efficiently and facilitate your work, openFT provides

- the partner list, see section “Setting up and administering the partner list” on page 66
- the **Transport Name Service** (TNS)

  openFT can only use TNS if CMX is installed and if operation with CMX and TNS has been enabled in the operating parameters (e.g. with *ftmodo -cmx=y -tns=y*).

  CMX makes both the TNS and functions for accessing the TNS available.

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.

Transport Name Service

Partner systems only have to be entered in the TNS if they are not connected via the TCP/IP transport system.

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

- You must explicitly activate the function in the operating parameters. To do this, you either enter the *ftmodo -tns=y* command or activate the *Use TNS* operating parameter option via the openFT Explorer.
- Operation with CMX must be enabled. If operation with CMX is disabled then TNS cannot be used.

For details, see section “Entering transport system applications in the TNS” on page 419.

3.8.1 Partner types

The partner list plays an important role during the administration f partners. A distinction is made between three types of partner system depending on whether and in what form partner systems are entered in the partner list.

- Named partners:
  All partners that are entered with their names in the partner list
- Registered dynamic partners:
  All partners that are entered without a name in the partner list
- Free dynamic partners:
  All partners that are not entered in the partner list

Registered dynamic partners and free dynamic partners are both simply referred to as dynamic partners.

Named partners

In FT requests, named partners are addressed using the names defined for them in the partner list.

You enter named partners in the partner list as follows:

`ftaddptn` *partner name* `-pa=`*address* `...`

These partners remain in the partner list until they are deleted from it using the *ftremptn* command. If authentication is required for the connection to a partner then this partner should be entered in the partner list.

The use of named partners has the following advantages:

– Complex partner addresses do not have to be specified explicitly in openFT commands.
– Security is enhanced because only partners that are genuinely recognized can be permitted.

Although a named partner can also be connected to via its address, in all openFT tasks such as logging or request queue activities, the partner name is displayed.

Registered dynamic partners

All partners that are entered only with their addresses but without names in the partner list are registered dynamic partners. They can only be accessed via the address and possess at least one attribute that differs from the default value for a free dynamic partner (see section “Free dynamic partners” on page 65).

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:
– Security level based on the partner setting (*-sl=p*) is the default setting for free dynamic partners and therefore does not count as a differently set attribute.
– In contrast, security level based on the operating parameter setting (*-sl=*; without parameters, default setting for the *ftaddptn* command) is a differently set attribute.

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

Free dynamic partners

Free dynamic partners are all the partners that are not entered in the partner list. They are therefore not displayed when you enter *ftshwptn* without specifying a partner name or partner address.

Partners of this type can only be connected to via their address and, with the exception of the security level (*-sl*), possess default attributes as described in the *ftaddptn* command. The security level for free dynamic partners is *-sl=p* (and not *-sl=* without parameters).

For the meaning of these attributes, see the *ftaddptn* or *ftmodptn* commands.

You can use the *ftmodptn* command to transform a free dynamic partner into a registered dynamic partner:

```
ftmodptn address ... (other options)
```

Enter a partner address that does not refer to any existing partner list entry and define one or more attributes with values other than the default (see above).

The advantage of the free dynamic partner concept is that users can address any required partners that are not entered in the partner list. This reduces the administrator's workload in terms of administration requirements. The disadvantage lies in the increased security risk and is the reason why you are also able to prohibit the use of dynamic partners, see page 66.

If the state of a free dynamic partner changes (e.g. to NOCON = Partner not available) and is therefore different from the default value then it is displayed in the partner list. However, it becomes a free dynamic partner again as soon as it once more becomes accessible (ACTIVE status).

Activating/deactivating dynamic partners

As system administrator, you may also prohibit the use of dynamic partners for security reasons. To do this, enter the following command:

```
ftmodo –dp=f
```

In this case, it is necessary to address partners via their names in the partner list. They cannot be addressed directly via their address. Inbound access is then also only permitted to partners that are entered with a partner name in the partner list.

You use *ftmodo -dp=n* to permit the use of dynamic partners again.

This function is also available in the openFT Explorer: *Administration* menu, *Operating Parameters* command, *General* tab.

3.8.2 Setting up and administering the partner list

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
– *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

0You can also administer the partner list via 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 the following context menu commands in the *Partner List* object window:
  – *New Partner List Entry...*: Enter a new partner
  – *Delete*: Delete partner
  – *Attributes*: Change the attributes of a partner.

For further details, refer to the online help system.

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

– 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
```

3.8.3 Specifying partner addresses

A partner address has the following structure:

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

host (= computer name), see page 68. This specification is mandatory; all other specifications are optional. In many cases, the other specifications are covered by the default values, so that the host name suffices as the partner address, see "Examples" on page 71. Final '.' or ':' can be omitted.

The individual components of the address have the following meanings:

protocol://

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

openft openFT partner, i.e. communication takes place over the openFT protocol.

ftam FTAM partner, i.e. communication takes place over the FTAM protocol.

ftp FTP partner, i.e. communication takes place over the FTP protocol.

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

Default value: **openft**

Exception: if a global name from the TNS is used for *host* and a presentation selector is assigned to this name in the TNS then **ftam** is the default value.

host

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

- Internet host name (e.g. DNS name), length 1 to 80 characters
- Global name from the Transport Name Service (TNS), up to 78 characters long, with full support for the 5 name parts. In this event, the following applies:
  - TNS must be activated (*ftmodo -tns=y*) and operation with CMX must be enabled to allow a global name from the TNS to be used in requests. In this case, the TNS name takes precedence over the Internet host name.
  - The partner address must end with *host* and must not contain any other address components, such as *port*, *tsel* etc.
  - *ftp* is not permitted for *protocol*, as openFT-FTP does not support TNS operation.
  - If the TNS entry contains a presentation selector for this global name, only *ftam* is permitted for *protocol*.
  - If the TNS entry does not contain a presentation selector, *ftam* is not permitted for *protocol*.

– IPv4 address with the prefix %ip, e.g. %ip139.22.33.44
You should always specify the IP address with the prefix %ip since the specification is then immediately treated as the IP address. Omitting this prefix results in performance impairments since in this case a search is initially performed in the TNS and then in the file /etc/hosts.
The IP address must always be specified as a sequence of decimal numbers separated by dots and without leading zeros.

– IPv6 address with the prefix %ip6, e.g.
`%ip6[FEDC:BA98:7654:3210:FEDC:BA98:7654:3210]` (IPv6) or
`%ip6[FE80::20C:29ff:fe22:b670%5]` (IPv6 with Scope ID)

The square brackets [..] must be specified.

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

port

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

Default value: **1100** for openFT partners.
A different default value can also be set in the operating parameters using *ftmodo -ftstd=*.

4800 for FTAM partners.

21 for FTP partners

11000 for ADM partners

tsel

Transport selector under which the file transfer application is available in the partner system. The transport selector is only relevant for openFT and FTAM partners.
You can specify the selector in printable or hexadecimal format (0xnnnn...). The specification will depend on the type of partner:

– openFT partner:
Length, 1 through 8 characters; alphanumeric characters and the special characters # @ $ are permitted. A printable selector will be coded in EBCDIC in the protocol and may be padded with spaces internally to the length of eight characters.

Default value: **$FJAM**

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

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

Default value: **$FTAM**

Note:
As a rule, **SNI-FTAM** must be specified for Windows partners with openFT-FTAM up to V10. As of openFT-FTAM V11 for Windows, the default value has been changed to **$FTAM** and can therefore be omitted.

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

ssel
Session selector under which the file transfer application is accessible in the partner system. You can specify the selector in printable or hexadecimal format (0xnnnn...). Length, 1 through 10 characters; alphanumeric characters and the special characters @ $ # _ - + = * are permitted. A printable selector is encoded in ASCII with a variable length in the log.

Default value: empty

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

psel
Only relevant for FTAM partners.
Presentation selector under which the file transfer application is available in the partner system. You can specify the selector in printable or hexadecimal format (0xnnnn...). Length, 1 through 10 characters; alphanumeric characters and the special characters @ $ # _ - + = * are permitted. A printable selector is interpreted as ASCII with a variable length in the log.

Default value: empty

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

Examples

The partner computer with the host name FILESERV is to be addressed over different protocols/connection types:

Connection type/protocol	**Address specification**
openFT partner	FILESERV
FTAM partner (Windows system as of V11.0, BS2000 or Unix system with default setting as of V11.0)	ftam://FILESERV
FTAM partner (Windows system with default setting up to V10.0)	ftam://FILESERV:.SNI-FTAM
Third-party FTAM partner	ftam : / /FILESERV:102.TS0001.SES1.PSFTAM
FTP partner	ftp://FILESERV

3.8.4 FTAC security levels for partner entries

If the FTAC functionality is to be used, the FT administrator should coordinate with the FTAC administrator to additionally define the security level relevant to FTAC for each partner in the partner list. To do this, the FT administrator uses the *-sl* option in the *ftaddptn* or *ftmodptn* command. Alternatively, in the openFT Explorer: Use the options of the *Security Level* group in the *Partner List Entry* dialog box.

The security levels regulate the degree of protection with respect to the partner system. A high security level is used when a high degree of security is required, and a low level for a low degree of security. When FTAC is first used, the security levels should be assigned in multiples of ten. This leaves the option open to incorporate new partner systems flexibly into the existing hierarchy.

If the degree of required security changes with respect to a partner system, the security level of the partner system can be modified with the command *ftmodptn* to meet the new requirements.

You can also use the operand *-sl=p* in the *ftaddptn* and *ftmodptn* command to activate the following automatic mechanisms for the security levels:

- Partners that are authenticated by openFT are assigned security level 10.
- Partners that are known in the transport system are assigned security level 90.
- Partners which are only accessed via their IP address (e.g. FTP partners) are assigned security level 100.

This automatic mechanism can be activated on a partner-specific basis (*ftaddptn* and *ftmodptn*) or globally by means of *ftmodo*.

This automatic mechanism also applies to all partners that are not entered in the partner list (free dynamic partners) irrespective of the settings made in the operating parameters..

If no security level was specified when a partner was generated (with *ftaddptn* or using the openFT Explorer), then openFT uses the global settings in the operating parameters (*ftmodo*). Here, it is also possible to specify a fixed security level as the default.

The security level of a partner entry is taken into account when a user wants to process a request via this partner entry. FTAC compares the security level of the partner entry with the security level for this function (e.g. inbound sending) specified in the user's admission set. If the security level in the admission set is lower than that in the partner entry, the request is rejected by FTAC. If a privileged FTAC profile is used for the request, the user can override the restrictions defined in the admission set.

3.8.5 Outbound and inbound deactivation of named partners

You are able to deactivate specific named partners for asynchronous outbound requests or for inbound requests.

In the case of outbound requests, you can also enable automatic deactivation which deactivates the partner for outbound requests after five unsuccessful attempts to establish a connection. Before any further attempt to establish a connection is possible, this partner system must be activated again manually. This prevents costs from being incurred unnecessarily since, under certain circumstances, even unsuccessful attempts to establish a connection may be charged for.

You can assign these settings either with the *ftaddptn* command when setting up the partner system or subsequently by means of the *ftmodptn* command.

3.8.6 Serialization of asynchronous outbound requests

You can force the serialization of asynchronous outbound requests for a partner system. You do this in the *ftaddptn* and *ftmodptn* commands by specifying the *-rqp=s* option or by activating the option *Serialized Processing of Asynchronous Outbound Requests* in the openFT Explorer.

This prevents the "overtaking" effects that can arise when requests are processed in parallel. The following points apply to serial processing:

- A follow-up request is not started until the preceding request has terminated.
- Serialization includes preprocessing and postprocessing operations but not follow-up processing operations because these are independent of the request.

This function can be used, for example, in a branch-head office configuration in which the branches send multiple files to the head office at the same time (daily, weekly or monthly figures). If serialization is enabled for the partner "head office" in the branch computers then each branch computer can only have only one active connection to the head office computer at any one time. This prevents bottlenecks at the head office computer of the sort that occur, for example, if the connection limit is regularly exceeded (see also the CONN-LIM parameter in *ftshwo*).

3.9 Monitoring with openFT

openFT provides the option of monitoring and displaying a range of characteristic data for openFT operation. The data falls into three categories:

- Throughput, e.g. total network throughput caused by openFT
- Duration, e.g. processing time for asynchronous jobs
- State, e.g. number of requests currently queued

You must be an FT administrator in order to activate, deactivate or configure monitoring.

If the asynchronous openFT server has been started and monitoring is activated (*ftmodo*), any user can call up the data and display it on the basis of certain criteria (*ftshwm*).

3.9.1 Configuring monitoring

You use the *ftmodo* command (page 229) 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.

3.9.2 Displaying monitoring data

You can call up the monitoring data at any time provided that monitoring is activated and the asynchronous openFT server is started. You can output the data via the into following ways:

- using the command *ftshwm*.
- using the openFT Monitor
- using preprocessing

3.9.2.1 Displaying local monitoring data using the ftshwm command

ftshwm outputs the monitoring data in the form of tables that you can further process as required either programmatically or using an editor.

When you call *ftshwm*, you specify what monitoring data is to be output, the format in which it is to be output (formatted, raw, tabular, or in CSV format), and the interval at which output is to be updated.

You will find details on *ftshwm* on page 322.

3.9.2.2 Displaying local or remote monitoring data via the openFT Monitor

A graphics-capable terminal is required for output with the openFT Monitor. By default, the openFT Monitor outputs the data in the form of one or more charts. The charts show the current state and history of the monitoring data. You can set what values are to be displayed in the openFT Monitor and store the setting for subsequent sessions. It is also possible to display all the monitoring data in tabular format in a graphics window.

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 (see page 274). When you start the program, you also specify the interval at which output is to be updated. For further details on the openFT Monitor, refer to the online Help system.

Displaying remote monitoring data via the openFT Monitor

The openFT Monitor allows you to view the monitoring data of openFT instances on the other systems. In order to do this, 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.

3.9.2.3 Displaying monitoring data via preprocessing

You can restrict access from a remote system to the transfer of monitoring data. In order to do this, you define an admission profile by specifying a file name prefix with the keyword *FTMONITOR as a preprocessing command. *FTMONITOR is a keyword for openFT that causes monitoring data to be transferred in the form required by the graphical openFT Monitor.

You can also view monitoring data from other systems as line-based output. You do this by using the file transfer commands *ft* and *ncopy* in combination with an admission profile that contains the preprocessing command *FTMONITOR. .

Example

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 line-based 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)
   ```

   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 line-based 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.

3.10 Security in FT operation

A user wanting to access resources of a system must always provide the system with proof of his or her authorization for the access. In the case of file transfer activities, access authorization must be verified in both the remote system. Verification usually entails specifying a user ID and a corresponding password.

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

- Authentication
- Encryption during data transfer, see page 86
- Using the FTAC function, see page 91

In addition, openFT provides an extended sender verification function (see page 85) that can be used, for example, if it is not possible to work with authentication, as well as mechanisms that protect against file inconsistencies (see page 87).

3.10.1 Authentication

If data requiring an extremely high degree of security is to be transferred, it is important that the respective partner system undergo a reliable identity check (“authentication”) before the transfer. The two openFT instances that are engaged in a transfer must be able to mutually check each other using cryptographic means, to ensure that they are connected to the “correct” partner instance.

In versions of openFT after version 8.1, for Unix systems and Windows systems or version 9.0 for BS2000 and z/OS, an expanded addressing and authentication concept is supported. This is based on the addressing of the openFT instances, using a network-wide, unique ID, and the exchange of partner-specific key information.

When communicating with partners that are using openFT version 8.0 (or older), the functions described in the following are not usable. The previous addressing concept is still supported for these partners for the sake of compatibility. In FTAM partners, authentication is not available in this form, since the FTAM protocol standardized by the ISO does not provide for comparable functionality.

3.10.1.1 Authentication usages

The following basic rule applies to mutual authentication: The instance that wants to authenticate another instance must possess the latter's public key.

Basically, there are three distinct usages:

- Case 1:
  For the local openFT instance, it is important that the supplied data comes from a secure source.

  To ensure this, the local openFT instance checks the identity of the partner instance. This assumes that a current, public key of the partner instance was stored locally, see section “Administering the keys of partner systems” on page 83.

  A configuration of this kind makes sense, for example, if a server’s files are to be accessed via openFT. It is important for the local openFT instance, that the received data come from a reliable source (the authenticated partner). In contrast, the source of an access attempt is unimportant to the server.

- Case 2:
  For the partner system, it is important that only a secure local openFT Instance is able to access its data.

  To ensure this, the partner instance checks the identity of the local openFT instance. This requires that a current, public key of the local openFT instance is stored in the partner instance (re-coded for BS2000- and z/OS- or OS/390 partners), see section “Distributing the keys to partner systems” on page 84.

  A configuration of this kind would be conceivable, for example, if partner systems in several branch offices were to be accessed from a central computer via openFT and the branch computers were only permitted to access the central computer (and, in fact, only the central computer).

- Case 3:
  For both the local openFT instance and the partner instance, it is important that the data comes from a reliable source and ends up in safe hands (combination of case 1 and case 2).

  To ensure this, both instances check the identity of the reciprocating system. For this to be possible, both systems must have exchanged a current public key and stored this in the partner instance.

3.10.1.2 Instance Identifications

The instance ID is a unique name up to 64 characters long, that must be unique throughout the network irrespective of case. It is particularly important when authentication is used.

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

Modifying the local instance ID

An instance ID may consist of alphanumeric characters as well as special characters. It is advisable to use only the special characters ".", "-", ":" and "%". The first character must be alphanumeric or be the special character “%”. The character “%” can only be used as an initial character. An alphanumeric character must follow a “.”.

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

- If the openFT instance has a network address with a **DNS name** you should use this as the ID. You can create an “artificial” DNS name for an openFT instance, by placing another part of a name in front of an existing “neighboring” DNS name, separated by a period.
- If the openFT instance does not have a DNS name, but is connected to a TCP/IP network, you should use the following ID.
  - IPv4: **%ip***n.n.n.n* (*n.n.n.n* is the IPv4 address of the local openFT instance without leading zeros in the address components).
  - IPv6: **%ip6[***x*:*x*:*x*:*x*:*x*:*x*:*x*:*x***]** (without scope ID) or
    IPv6: **%ip6[***x*:*x*:*x*:*x*:*x*:*x*:*x*:*x***%***s***]** (with Scope ID)
    where *x:x:x:x:x:x:x:x* is the IPv6 address of the local openFT instance and *s* is the scope ID of the local network card.

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 */etc/hosts* or in the TNS. When TNS is used the global name must correspond to the instance ID of the partner.

With the aid of the instance IDs of the partner systems, openFT administers operational resources like, for example, request waiting queues and cryptographic keys.

3.10.1.3 Creating and administering local RSA key pairs

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

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

- *ftcrek* (or the openFT Explorer) creates RSA key pairs for the local openFT instance that currently consist of a private key and a public key.
- *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

A RSA key pair set in the Unix system currently consists of three key pairs with a lengths of 768, 1024 and 2048 bits.

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 "Instance directory" on page 26) 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:

– BS2000/OSD: Value of the system parameter HOSTCODE
– z/OS: IBM1047
– 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*.

3.10.1.4 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 "Key pair attributes" on page 81. 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" on page 83.

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

openFT supports key files in the following formats:

- PEM format (native PEM)

  The PEM-coded files must be present in EBCDIC format.
- PKCS#8 format encrypted without password phrase or after v1/v2 with password phrase (PEM-coded).

  You must specify the password phrase used for encryption in the password parameter when you perform the import.
- PKCS#12 v1 format in the form of a binary file. The file is searched for a private key and any non-supported elements (e.g. certificates, CRLs) are ignored during the import. If the certificate is protected by a signature or hash then openFT does not perform a validity check. The validity of the file must be verified using other means. The first private key that is found in the file is imported. Any others are ignored.

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

3.10.1.5 Administering the keys of partner systems

The public keys of the partner systems have to be stored in Unix systems as files in the directory *syskey* of the instance file tree of the local openFT instance, see “Instance directory” on page 26 (Standard: */var/openFT/std7syskey*). 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. 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 “Note on Secure FTP” on page 87.

Modifying the keys of partner systems

You can use the *ftmodk* command to modify the keys of partner systems by specifying an expiration date or modifying the authentication level (1 or 2):

- If you specify an expiration date then it is no longer possible to use the key once this date has expired.
- If you set authentication level 2 then openFT also performs internal checks. Level 2 is supported for all openFT partners as of Version 11.0B. Level 1 authentication attempts to this partner are rejected.

You can make these settings for a specific partner or for all partners, as you require, and modify them subsequently if necessary.

3.10.1.6 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 these files are re-coded (e.g. by transferring them as text files via openFT), if you transfer them to a partner with BS2000/OSD or z/OS (or OS/390) or Windows.

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

- For partners using openFT for 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 for Unix systems in the directory *syskey* of the instance file tree, see 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 for Windows in the directory *syskey* of the instance file tree. In the case of the standard instance the path name under Windows 7 is *%ProgramData%\Fujitsu Technology Solutions\openFT\var\std\syskey*.
  On older openFT versions with Windows XP, the path name is *openFT installation directory\var\std\syskey*.

  The instance ID of your local openFT instance must be selected as the file name.
- For partners using openFT for z/OS or OS/390 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.

3.10.2 Extended authentication check

openFT partners using openFT from version 8.1 onwards, support the authentication mechanism (see page 77). If the local system has a public key of the partner at its disposal, the partner's identity is checked by cryptographic means.

For partner systems that do not work with authentication, inbound requests are checked with the aid of the instance identification in order to ascertain whether the calling system has a valid entry in the partner list. openFT offers via sender checking the possibility of checking not only the processor name, but also the transport address.

The extended sender checking can be globally enabled for openFT partners or just for specific partners:

- globally, using *ftmodo -ptc=a*
  or via the openFT Explorer using *Operating Parameters* - *General*, *Partner Check*
- partner specifically, using *ftaddptn* / *ftmodptn -ptc=a*
  or via the openFT Explorer using the *Partner Check* section in the *Partner List Entry* dialog

The global setting applies to all partners for which partner checks are set as default (FTOPT output in *ftshwptn*).

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

Extended sender verification is of no relevance for dynamic partners because these are always identified via the transport address.

If the authentication check returns a negative result, the request is rejected.

3.10.3 Encryption on data transfer

openFT allows you to encrypt request description data and file contents.

If you want to use encryption for user data in addition to request description data, you must install openFT-CR V12.0. openFT-CR must also be installed on the partner system. For legal reasons, openFT-CR is not available in all countries.

If possible, openFT uses the RSA/AES procedure with a key length of 256 bits for encryption. In the case of connections with older partners, 128-bit RSA/AES or RSA/DES may also be used. The most secure procedure that is supported by both partners is always used

So that you can transfer openFT request description data and file content in encrypted form, there must be a RSA key pair set in the local system and encryption must not be deactivated (e.g. via specifying *ftmodo -kl=0*). You can check this using the *ftshwo* command. The output parameter KEY-LEN displays the length of the currently used RSA key in bits (0, 768, 1024 or 2048). 0 means that encryption is deactivated. You can set the length required for the RSA key via the operating parameters. To do this, use the option *-kl* in the *ftmodo* command or the openFT Explorer (*Administration* menu, *Operating Parameters* command). The default value after a new installation is 2048.

An RSA key pair set is created during new installation of openFT and consists of private and public keys of suitable length. Other key pair sets can be created (if necessary) using *ftcrek* or imported using *ftimpk*. Obsolete key pair sets are deleted using *ftdelk*. For further details on local keys, see section “Creating and administering local RSA key pairs” on page 80.

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.

In addition, the following applies:

- If outbound encryption is activated then the file content is encrypted on outbound requests even if no encryption is demanded in the request itself. If the partner does not support encryption (e.g. because it is deactivated or because openFT-CR is not installed) then the request is rejected.
- If an unencrypted inbound request is to be processed while inbound encryption is activated, then this request is rejected.

3.10.4 Protection mechanisms against data manipulation

openFT implicitly checks the integrity of the transferred data by communicating with openFT partners version V8.1 and later. The scope is defined in the transfer request:

– In the case of requests with encryption, the transferred file content is also checked.

– In the case of requests without encryption, an integrity check of the file content can be activated explicitly. To do this, use the option *-di* in the *ft* or *ncopy* command or choose the option *Integrity Checked* in the *Transfer File - Options* dialog box in the openFT Explorer.

– If neither encryption nor the integrity check are activated then only the integrity of the request description data is checked.

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

3.10.5 Note on Secure FTP

A Secure FTP server makes its key and the certificate available to the openFT instance for encryption purposes. No mutual authentication is carried out.

openFT is able to exchange encrypted outbound file contents with a Secure FTP server if openFT-CR is installed on the openFT side and the FTP server supports the TLS (**T**ransport **L**ayer **S**ecurity) protocol. AES (Advanced Encryption Standard) is used as the encryption method.

In the inbound direction, openFT does not support encrypted file transfer over the FTP protocol.

If openFT requires encryption of the file content, but the FTP server does not support the TLS protocol, the request is rejected. If openFT does not require encryption of the file content, the request description data is only encrypted if the FTP server accepts the TLS protocol, otherwise the request description data is transferred in unencrypted form.

3.11 openFT logging

As an openFT or FTAC administrator, you may

- display records of all users
- switch log file and administer offline logging page 89
- modify log settings, see page 89
- save and delete the log records of all users, see.

Aa an ADM administrator, you may

- display records of all users (and therefore save in files)
- delete the log records of all users

Log files are stored under the *log* directory of the corresponding openFT instance. A log file has the following name:

`syslog.L`*yymmdd*`.L`*hhmmss*

Where:

yy = year, 2-digit.
mm = month, 2-digit.
dd = day, 2-digit.
hh = hour, 2-digit.
mm = minute, 2-digit.
ss = second, 2-digit.

The date and time designate the time (GMT) at which the log file was created. This suffix makes it possible to distinguish between the current and offline log files, see page 89.

Displaying log records

You can use the *ftshwl* command to view all log records in the system. Using the polling options 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 the cause of rejection or abort of the request by means of a 4-digit reason code. This column can also contain a positive acknowledgment to a request (reason code 0000). 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.

In the *Logging* object window, you can then execute the following and other functions:

– View details of a log record
– Define criteria for the log records that are to be displayed
– Delete log records

Switching log file and administering offline logging

You can change the log file using 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.

This change-over has the following benefits:

– Faster access to logging information due to smaller log files.
– Improved administration of log records through regular change-overs and back-ups of the offline log files, see page 90.
– Possibility of performing extensive searches in the offline logging information without affecting ongoing openFT operation.

You can also change the log file in the openFT Explorer (*Administration* - *Operating Parameters* - *Logging*). 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 set the scope of the logging functions and define the times and intervals for the automatic deletion of log records.

Setting the scope of logging

You can set the scope of logging, i.e. what log records are to be written in the openFT Explorer under *Administration* - *Operating Parameters* - *Logging* or using the *ftmodo* command (options *-lt*, *-lc* and *-la)*.

Following installation, full logging is set. You can set the scope of FT, FTAC and administration function logging differently.

Setting the automatic deletion of log records

You can set the intervals for the automatic deletion of log records in the openFT Explorer under *Administration* - *Operating Parameters* - *Logging* or by means of the *ftmodo* command (options *-ld, -lda, -ldd* and *-ldt*). 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.

Saving of log records in files and deleting log records

Basically, openFT writes an indefinite number of log records. However, if no more storage space is available on disk, FT requests are rejected. It is therefore essential to limit the number of log records that are written to the volume necessary, to monitor the log records at regular intervals and to delete log records that are no longer required or transfer them to external storage.

All log records may be deleted by the openFT administrator, the FTAC administrator and the ADM administrator. To do this, use the *ftdell* command. Alternatively, you can also delete log records in the openFT Explorer (Logging object window, context menu command *Delete Log Records*).

If you need continuous documentation over an extended period, you should therefore back up the log records of the current log file or the offline log file(s) (e.g. in a file on CD or DVD) from time to time. To do this, redirect the output of *ftshwl* to a file and then delete these log records or offline log files.

- 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*.

The benefit of this is, first, that the log records provide a complete documentation which can be maintained over long periods, and second, that the current log file does not become unnecessarily large, thus resulting in slower access performance.

Deleting log records causes the size of the log file to change since the storage space is immediately free upon deletion.

3.12 Administering the FTAC environment

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

3.12.1 Administering admission sets

As the FTAC administrator, you specify the standard admission set and can view, modify and delete the standard admission sets for all users in the system.

The FTAC administrator is also responsible for specifying the ADM administrator initially, by setting the ADM privilege in the admission set of the ADM administrator (see section “Defining the ADM administrator” on page 118).

Standard admission set

The standard admission set applies to all login names. The user can restrict this admission set further.

The user can override the entries in the standard admission set only,
– if you, as FTAC administrator, modify the admission set of the user accordingly,
– or if you set up a privileged FT profile.

Following an initial installation or preinstallation 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.

Displaying and modifying admission sets

Admission sets can be viewed 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.

You can also view admission sets in the openFT Explorer by clicking on the *Admission Sets* object. You will find a detailed description of each of the functions in the online help.

The settings in the admission set apply to all users initially. As the FTAC administrator, you can assign an individual admission set for each user in the system or modify an existing one. The *ftmoda* command is available for this purpose.

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 page 72).

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 page 72

– and for openFT partner to undergo a reliable identity check using cryptographic means, see section “Authentication” on page 77.

The use of a file name prefix in the FT 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*, *tftp* must not be active.

3.12.2 Administering admission profiles

As the FTAC administrator, you can create FT profiles for any user in the system and modify them. The FTAC administrator is the only person who can assign privileges to FT profiles.

Creating FT profiles

You can create FT profiles with the command *ftcrep*. If you also want to assign a transfer admission at the same time, you must either have FT administrator rights as the FTAC administrator or specify the password for the particular login name. If you do not have FT administrator rights or specify the password, the profile is created without a transfer admission; the user must then assign it later.

When you create the profile, you can also assign privileges.

You can also create admission profiles in the openFT Explorer by opening the *Admission Profiles* dialog window via the *File/New* menu item. You will find a detailed description of each of the functions in the online help.

Viewing and modifying FT profiles

You can use the *ftshwp* command to display the FT profiles of all users. The transfer admission of the profile is not output, i.e. your administrator privileges do not grant you access to files on remote systems.

You can also view the admission profiles in the openFT Explorer by clicking on the *Admission Profiles* object. You can also change admission profiles in the *Admission Profiles* dialog window. You will find a detailed description of each of the functions in the online help.

You can use the *ftmodp* command to make the following changes to an FT profile:

– assign or cancel privileges
– modify the transfer admission of an FT profile whose owner is a different user ID. In order to do this you must have FT administrator rights or you must know the password of the profile's owner.
– assign the profile to another login name

If the FTAC administrator does not possess FT administrator rights or also specifies the profile owner's password then the profile is locked after this type of change. The profile owner must acknowledge the modification by unlocking the profile, for example with the command *ftmodp ... -v=y*.

If a FT profile is private (*-u=pr*, see section “ftcrep - Create an FT profile” on page 187) and if a corresponding transfer admission is assigned for a second time, the existing transfer admission is locked.

Locking/unlocking FT profiles

There are two types of lock:

- The lock consists of a transfer admission that has been set to "invalid".

  There may be four different reasons for this:

  – The lock has been set explicitly with *-v=n*.

  – The FTAC administrator has changed the properties of a profile that does not belong to him/her.

  – The FTAC administrator has imported profiles that do not belong to him/her.

  – The transfer admission of a private profile (*-u=pr*) has been accidentally – or maliciously – "disclosed".

  To release the lock, set the transfer admission to valid again (*-v=y*).

- A profile has reached its expiration date.

  To release the lock, set an expiration date in the future or delete the expiration date.

Deleting FT profiles

You can use the *ftdelp* command to delete FT profiles of a user. 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 also delete admission profiles via the openFT Explorer by selecting the *Delete* command from the context menu. You will find a detailed description of the object windows in the online help.

Assigning privileges to FT profiles

A privileged FT profile is intended for exceptional circumstances in which it is necessary for a user to override all restrictions. To assign privileges to a profile, you can use the command *ftmodp ... -priv=y*, for example.

Once a profile has been assigned privileges, the owner of the profile can only modify the transfer admission and withdraw the privilege. To prevent abuse, no other changes are permitted.

You can also assign privileges to admission profiles via the openFT Explorer in the *Admission Profiles* dialog window. You will find a detailed description of each of the functions in the online help.

3.12.3 Saving the FTAC environment

When migrating individual users to another processor, or when migrating the complete processor, it is possible to provide the users with the same FTAC environment by saving the admission sets and FT profiles and restoring them on the new processor. Furthermore, you can also created backup copies of the FTAC environment on your processor by this method.

Saving admission sets and FT profiles

You can use the *ftexpe* command for backups. You can select the admission sets and FT profiles which you wish to save for particular users. You must specify the name of the backup file.

In all cases, the standard admission set is not included in the backup. Instead, all the values of an admission set that refer to the standard admission set (represented by an asterisk (*) in the display) are stored as variables. This means that when they are restored, they will receive the value of the standard admission set valid at the time.

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

Displaying saved admission sets and FT profiles

You can display saved admission sets and FT profiles with the *ftshwe* command. You must specify the name of the backup file.

You can also view saved admission sets and admission profiles via the openFT Explorer by dragging the export file into the *Exported Admissions* directory and then dropping it there or by selecting the *Open Export File* command in the context menu of the *Exported Admissions* object.

Importing saved admission sets and FT profiles

You can re-import saved admission sets and FT profiles with the *ftimpe* command. Here, you must 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.
You can also import admission sets and admission profiles via the openFT Explorer using the *Import FTAC Environment* command in the *Administration* menu. You will find a detailed description to it in the online help.

3.13 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 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 CMS 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.

*Open*FT 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 also chapter “Administering openFT via SNMP” on page 103).

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 the startup and shutdown files are initialized as during a new installation.

  When an existing instance is activated, 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 default instance *std*, You must assign the default 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 default 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. 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.

  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.

If you work with more than one instance, then in this case a separate *ftalarm* call is required for each instance (see also section “ftalarm - Report failed requests” on page 180).

You will find detailed descriptions of the *ftcrei*, *ftmodi*, *ftupdi* and *ftdeli* commands in chapter “openFT commands for the administrator” starting on page 157. The *ftseti* and *ftshwi* commands are described in the “openFT for Unix systems” User Guide.

Startup and shutdown file

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

3.14 Diagnosis

To support error diagnostics, you can switch a trace on or off, trace files and output diagnostic information. These functions are primarily intended for the Maintenance and Diagnostic Service of Fujitsu Technology Solutions.

Switching on and off trace mode

You can switch the trance mode on or off with the FT command *ftmodo* or via the openFT Explorer (dialog *Operating Parameters - Trace* from the *Administration* menu). When the trace mode is enabled, the diagnostic data is written to trace files, which must be edited for further evaluation.

Preparing trace files

The trace files are located in the directory *traces* of the respective openFT instance; see "Instance directory" on page 26. In the case of the standard instance the path name is */var/openFT/std/traces*.

The trace files can be displayed in the openFT Explorer:

– for the local instance using the *Open Trace File* command in the *Administration* menu

– for remotely administered instances, using the *Traces* object directory of the relevant instance.

Further possibility: In the openFT Explorer, navigate to the directory *traces*, and in the object window, open a trace file using the *View* command from the context menu. You will find a detailed description of each of the functions in the online help.

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.

Message file for console commands:

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; see “Instance directory” on page 26. 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.

3.15 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
  ```

  The file *partner_save* 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
  ```

  The file *option_save* 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 Administering openFT via SNMP

In order to administrate openFT via SNMP, your processor must be have a EMANATE master agent.

The openFT subagent is available for Solaris/Sparc and HP-UX platforms. It is supplied with openFT and is set up when openFT is installed.

4.1 Activities after installation

After installation of openFT, different activities are required.

1. If your system is not already being administered with SNMP, you will need to activate administration via SNMP.

   You will need a community string with write authorization to administer openFT via the openFT subagent. If you only have read authorization, 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 page 105).

   Consult your platform specific documentation to find out how to activate the SNMP administration.

2. Start the agent (see below)

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.

4.2 Starting the openFT subagent

There are the following two ways to start the openFT subagent:

- Enter `/opt/bin/ftagt &`.

  The openFT subagent is then started and remains active until the system is shutdown.

- On Solaris, you can automatically start the openFT subagent via SMF, see section “Solaris SMF” on page 41.

- On HP-UX, remove the comment symbol in the line of the startup file that contains the word *ftagt* (for example: */var/openFT/std/etcinit/openFTinst*) as well as in the corresponding line in the startup file of any other instances. The openFT subagent is then also started each time the system is booted.
  If you want to terminate the openFT subagent, then you can do this with a *kill -2* command with the process number of the openFT subagent as the parameter.

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” on page 96).

4.3 SNMP management for openFT

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 above-mentioned management tasks. It is located in the file */opt/openFT/snmp/openFT.asn1*.

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

4.3.1 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 stop	1 2	the asynchronous openFT server is started the asynchronous openFT server is stopped

Output

Syntax	Integer	Meaning
on off	3 4	the asynchronous openFT server is started 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").

4.3.2 System parameters

MIB definition

Object name/object identifier	Access	Meaning	Command *ftmodo*
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	*-tu*
ftSysparMaxOSP/1.3.6.1.4.1.231.2.18.2.7.0	read-write	Max OSP[1]	*-cl*
ftSysparMaxISP/1.3.6.1.4.1.231.2.18.2.8.0	read-write	Max ISP[1]	*-cl*
ftSysparProcessorName/ 1.3.6.1.4.1.231.2.18.2.9.0	read-write	Processor Name	*-p*
ftSysparStationName/ 1.3.6.1.4.1.231.2.18.2.10.0	read-write	Station Name	*-l*
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)	*-ccs*
ftSysparMaxInboundReqs/ 1.3.6.1.4.1.231.2.18.2.12.0	read-write	Max Inbound Requests	*-rql*
ftSysparMaxLifeTime/ 1.3.6.1.4.1.231.2.18.2.13.0	read-write	Max Life Time	*-rqt*

[1] The 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 starting on page 229.

4.3.3 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

4.3.4 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 on	1 18	Diagnosis management is deactivated 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).

4.3.5 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” on page 80.

5 Central administration

Central administration in openFT covers the functions **remote administration** and **ADM traps**. openFT for Unix systems provides full support for both functions.

Compared with openFT up to V10.0, these functions offer considerable advantages that are of particular benefit if you want to administer and monitor a large number of openFT instances. These benefits include:

- Simple configuration

  The configuration data is maintained centrally on the **remote administration server**, which means that it only exists once. The creation of roles in the form of **remote administrators** and the grouping of several instances make it possible to implement even complex configurations simply and in a clearly structured way. Subsequent changes are simple to incorporate and thus make the configuration easy to maintain.

  The remote administration server runs on either a Unix or a Windows system.

- Simplified authentication procedure

  If you wish to use authentication for reasons of security, it is only necessary to distribute a few keys:

  – For the direction to the remote administration server, the keys of computers from which administration is to be performed must be stored on the remote administration server.

  – For the direction from the remote administration server to the instances to be administered, it is only necessary to store the public key of the remote administration server on the openFT instances to be administered.

- High performance

  The remote administration interface allows far longer command sequences than in openFT up to V10.0.

  In addition, it is possible to configure the remote administration server in such a way that it is available exclusively for remote administration. In this case, there is no dependency on normal FT operation and hence no mutual impact.

- Simple administration

  Remote administrators only need one (central) transfer admission. Up to openFT V10, the remote administrators had to remember the access data for each openFT instance to be administered.

- Central logging of important events

  ADM traps can be generated if certain events occur on openFT instances. These are sent to the (central) ADM trap server and stored permanently there. This allows remote administrators to evaluate important events at a later time and for specific instances.

- Compatible integration of earlier openFT versions

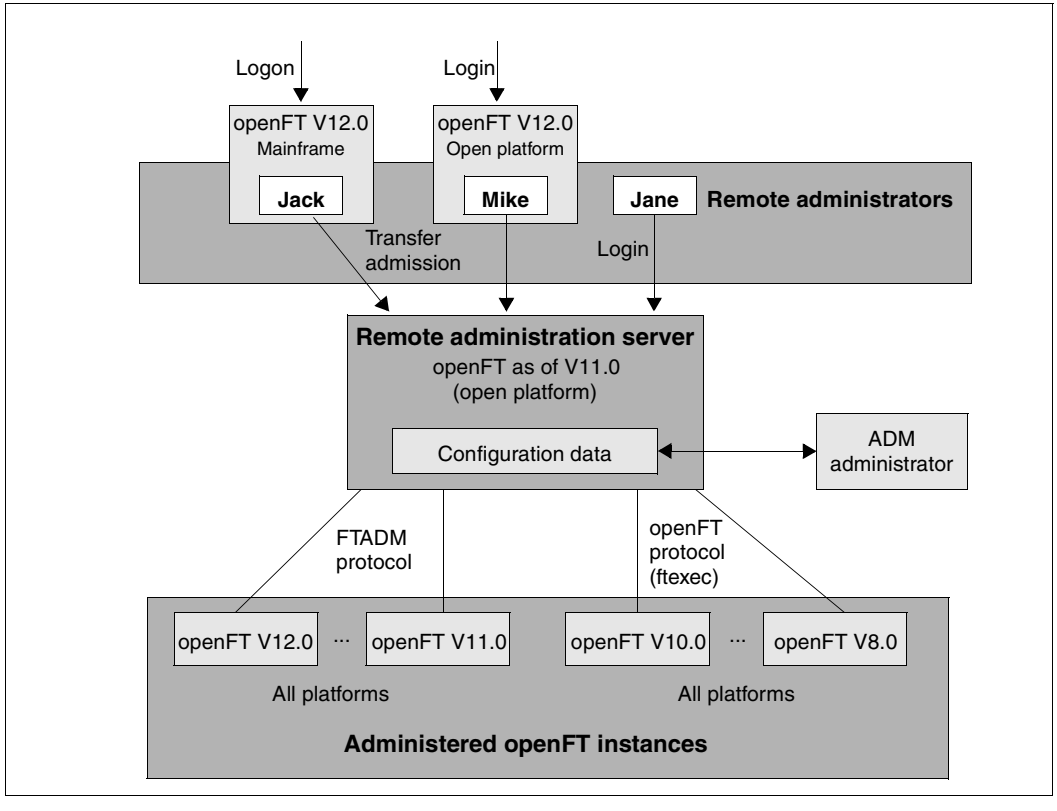
  Instances running versions of openFT as of V8.0 can simply be added to the configuration and administered in the same way as instances as of V11.0. All the administration functions offered by the corresponding openFT version can be used.

5.1 Remote administration

openFT allows you to set up a remote administration server via which you can administer your openFT instances on the various platforms. You can choose to use any openFT instance as an administration workstation.

5.1.1 The remote administration concept

The figure below shows the remote administration components and the most important configuration options on the basis of a deployment scenario.



Remote administration components

Remote administration comprises the following components:

Remote administration server

Central remote administration component. This runs on a Unix or Windows system with openFT as of V11.0 and contains all configuration data for remote administration.

Multiple remote administration servers can be defined in a complete configuration. See page 115.

ADM administrator

Person who administers the remote administration server. This person creates the configuration data for remote administration in which, for instance, the remote administrators and the administered openFT instances are defined. The ADM administrator is the only person permitted to change the configuration data.

Remote administrator

Role configured on the remote administration server and which grants permission to execute certain administration functions on certain openFT instances. A remote administrator can

– Log in directly at the remote administration server (single sign-on)

– log in to a different openFT instance (as of V11.0) and access the remote administration server using an FTAC transfer admission.
The openFT instance can be running either on a mainframe (BS2000/OSD, z/OS) or on a Unix or Windows system. The FTADM protocol is used for communication.

Several remote administrators can be configured with different permissions.

Administered openFT instance

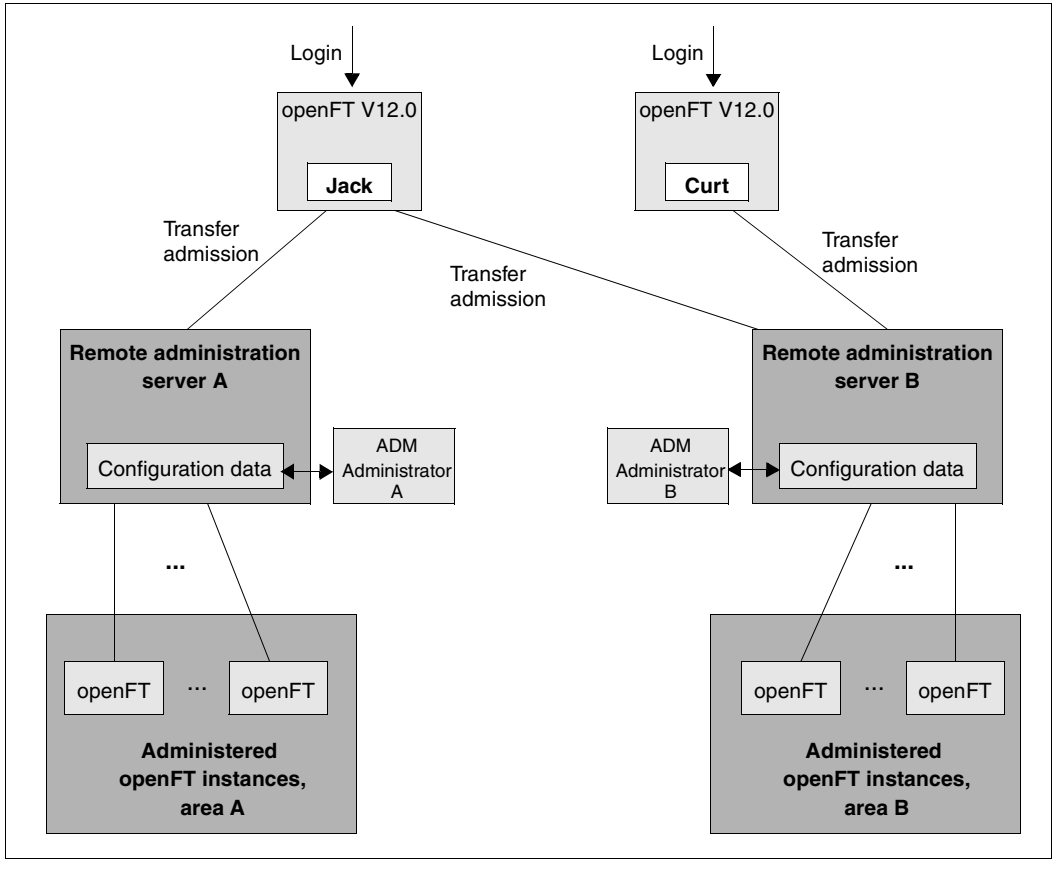
openFT instance that is able to be administered by remote administrators during live operation. Access is via an admission profile. The following applies, depending on the openFT version of the openFT instance:

– In the case of openFT instances as of V11.0, the FTADM protocol is used, and the full range of remote administration functions can be utilized.

– In the case of openFT instances from V8.0 through V10.0, administration is carried out using the openFT protocol and the command *ftexec*. The range of functions available depends on the openFT version of the instance being administered.

Configuration with multiple remote administration servers

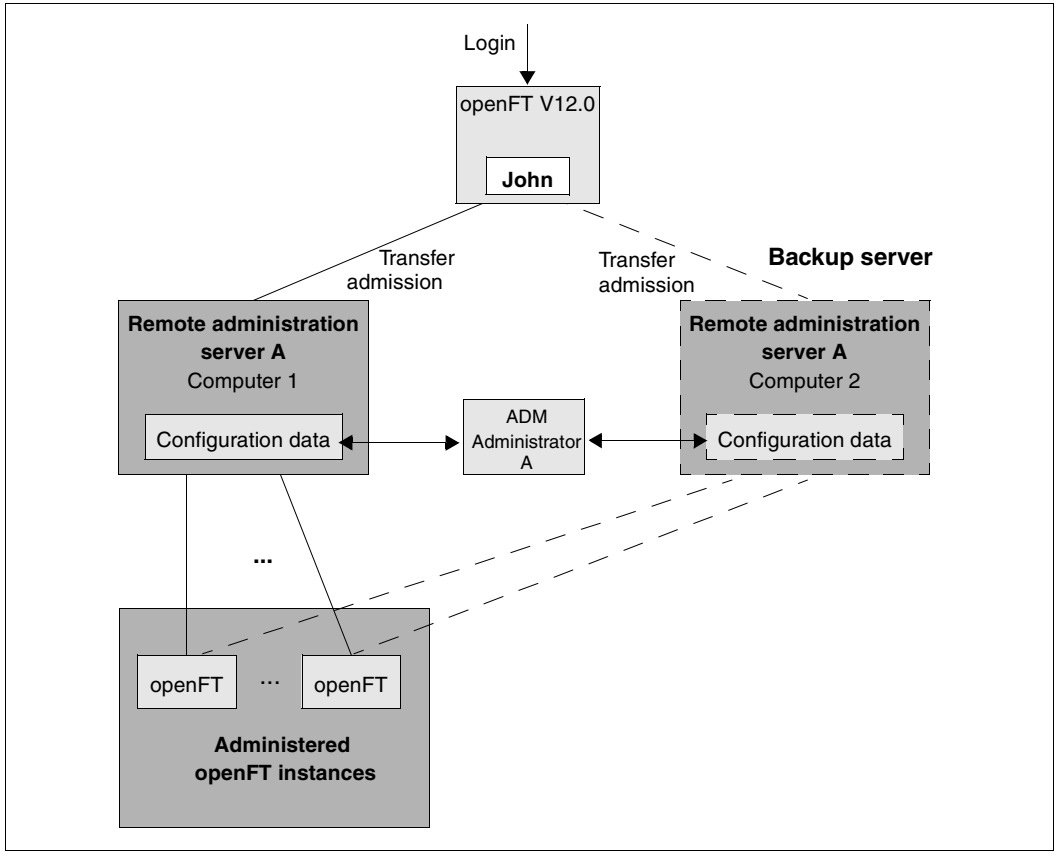
Complex configurations can also be defined in which remote administrators access multiple remote administration servers. The figure below shows an example of this.



Separate configuration with two remote administration servers

Areas A and B are theoretically strictly separated, but *John* is permitted to administer instances from areas A and B, whereas *Curt* can only administer instances from area B.

The same method can also be used to define a redundant configuration with a second remote administration server. This allows implementation of a simple backup solution.



Redundant configuration with a second administration server as a backup.

If Computer 1 fails, the remote administrator can use Computer 2 as the remote administration server. In order to do this,

– the ADM administrator must always ensure that the configuration data on the two computers is consistent,
– the admission profiles for accessing the remote administration server and the partner list entries (if they are used) are identical on Computer 1 and Computer 2,
– the admission profiles on the administered instances are defined in such a way that they accept both remote administration servers as partners.

If authentication is used, you must also note that

– the keys for the computers from which administration is performed must be present on both remote administration servers,

– the administered instances require the keys of both remote administration servers.

For this reason, with complex configurations in particular, you should implement failsafe protection of the remote administration server using a cluster. You can find examples of how to set up a cluster in the section “openFT in a Cluster with Unix based systems” on page 428.

5.1.2 Configuring the remote administration server

The remote administration server stores the data required for remote administration and 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 V12.0.

The following table indicates

- the steps required to create a configuration as shown on page 113,
- 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 partner list	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 page 136.

It now remains to configure openFT instances on the partner systems for remote administration. See page 138.

5.1.2.1 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 FTAM 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. 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*.

5.1.2.2 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.

– 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 page 147.

– 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.

5.1.2.3 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 page 126.

transfer-admission
: Identifies the FTAC transfer admission. The remote administrator must specify this with a remote administration request. See page 141.

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*.

5.1.2.4 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*. See the section “ftaddptn - Enter a partner in the partner list” on page 167. 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.

  The ADM administrator must additionally specify the attribute *Mode="Legacy"* in the configuration file for such partners. See the section “Defining instances” on page 129ff.

5.1.2.5 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” on page 136.

The representation of the configuration corresponds to the display you will subsequently see under *Remote Administration* in the openFT Explorer, see the example under “Modifying the configuration file” on page 123.

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.

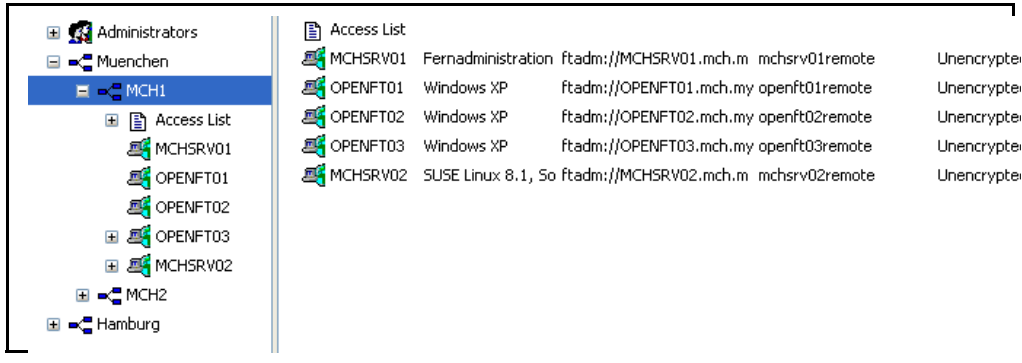
  You can use the *Properties* command in an access list's context menu to open the *Access List* dialog box. Here you can specify whether access permissions are to be inherited from parent access lists. This dialog box also displays any access permissions that may have been inherited.

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*.

5.1.2.6 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 Using the XML template and XML schema.

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 page 125,
- the remote administrators, see page 126,
- the openFT instances and groups of instances to be administered by these remote administrators, see page 128,
- the remote administration permissions that the remote administrators have on each of the openFT instances (access list), see page 132.

The ADM administrator must then import the configuration file into the remote administration server using the *ftimpc* command. See page 136. The *ftexpc* command (see page 209) 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” on page 151.

Using the XML template and XML schema

The directory *samples/ftadm* 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 V12.0, `"1100"` 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="1100"
   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 “Defining remote administrators” on page 126.
- 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 “Defining an access list” on page 132.
- 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 on page 126. For a detailed description of a group, refer to the section “Defining groups and openFT instances to be administered” on page 128.
- 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 the section “Defining groups and openFT instances to be administered” on page 128.

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 page 142.

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” on page 119.

Example:

```
<Configuration
    Version="1100">
    <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 the section “Defining an access list” on page 132.
- 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 on page 126.
- 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.

Specification of the *group* and *instance* elements within a group is optional, but a group must contain a 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>
            <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 on page 132.

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” on page 138:
  - openFT as of V11.0:
    REMOTE-ADMINISTRATION (corresponds to *ftcrep ... -ff=a*)
  - openFT < V11.0:
    TRANSFER-FILE + FILE-PROCESSING (corresponds to *ftcrep ... -ff=tp*)

  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 non-standard 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 "Defining an access list".

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 XP"
         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.

Defining 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 page 132.

And 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 page 126. 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 remote administrator is permitted 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 9"
       Address="HHWSRV11.hhw.mycompany.net"
       Admission="hhwsrv11remote"
       Mode="Legacy">
       <AccessList>
          <AccessEntry
             AdministratorID="Mike"
             DenyFunction="FTAC"/>
       </AccessList>
    </Instance>
</Group>
```

5.1.2.7 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 page 124.

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.

5.1.2.8 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.

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.
For further details, see the section “ftshwc - Show openFT instances that can be remotely administered” on page 288.

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 page 124 ff.
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” on page 121 or section “Creating a configuration file using a text or XML editor” on page 124.
3. Import the changed file, e.g. using *ftimpc xml-file*. See also page 136.

   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.

5.1.3 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 page 129.

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.

5.1.3.1 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/OSD:

```
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/OSD and z/OS).

5.1.3.2 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 page 138).

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/OSD:

```
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)
```

5.1.4 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 page 143).

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 "Determining the names of the openFT instances".

5.1.4.1 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 page 119) and the profile name must be assigned to a remote administrator in the configuration file (see page 126).

Sample output

```
TYPE    = *INSTANCE     ACCESS = FT+FTOP+FTAC
   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 are listed under ACCESS. See also the description of *ftshwc* on page 288.

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 "Defining remote administrators" on page 126.

  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 page 141.

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. For further details, see "ftadm - Execute remote administration command" on page 172.

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 page 119) and the profile name must be assigned to a remote administrator in the configuration file (see page 126).

5.1.4.2 Remote administration using the openFT Explorer

The object tree of the openFT Explorer contains the item *Remote Administration* with the following icon:

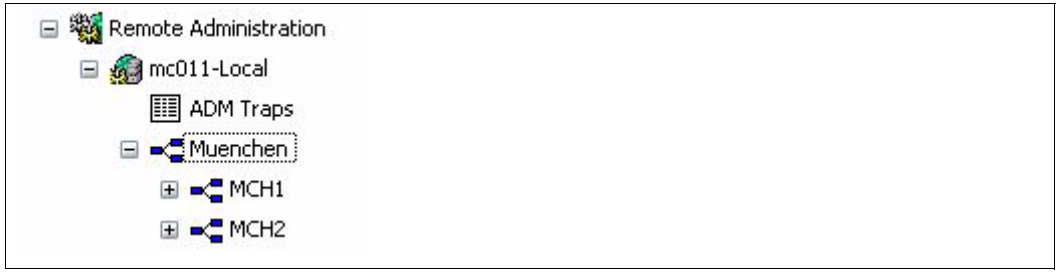
You can log in to the remote administration server locally or perform remote administration from a remote computer.

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 page 119) and the profile name must be assigned to a remote administrator in the configuration file (see page 126).

     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 page 143).

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 *London*). 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.

5.1.5 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, see the section “ftshwl - Display log records and offline log files” on page 297,
- and you can delete ADM log records with the *ftdell* command provided that you have the appropriate permission, see the section “ftdell - Delete log record or offline log file” on page 204.

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

ADM traps are short messages that openFT sends to the **ADM trap server** if certain events occur during operation of openFT. Such events may include errored FT requests, status changes or the unavailability of partners, for instance.

The ADM traps are stored permanently on the ADM trap server. This allows openFT systems to be monitored at a central location. The FT administrator of the ADM trap server is thus provided with a simple way of gaining an overview of events that have occurred on the openFT instances he is monitoring using the openFT Explorer or the *ftshwatp* command.

If the ADM trap server is simultaneously used as a remote administration server, remote administrators can also view traps from other systems and hence monitor the systems that they are administering.

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 page 149.

- 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* 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 default instance, the pathname is */var/openFT/std/log/sysatpf*.

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/OSD and z/OS systems in the relevant openFT "Installation and Administration" manuals.

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 page 147.
- 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

  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:

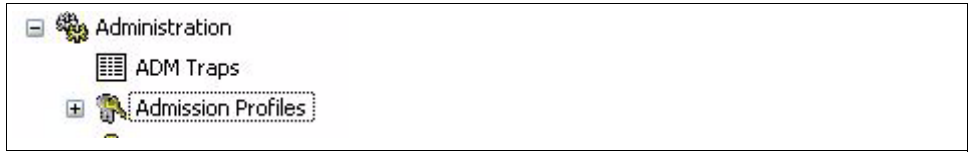
1. Choose *Administration*, *Operating Parameters...* to open the *Traps* tab.
2. In the *ADM Trap Server* group, enter the name of the ADM trap server and the transfer admission.
3. In the *ADM* column of the *Type* group, select the events on which ADM traps are to be sent.

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.). For details, see the section “ftshwatp - Display ADM traps” on page 281.
  - 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 “Determining the names of the openFT instances” on page 141. 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, see page 281. 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 page 119) and the profile name must be assigned to a remote administrator in the configuration file (see page 126).

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	**Sub-group**	**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="1100"
   Description="Configuration for central server MCHSRV01">
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="/opt/openFT/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="ProfileO2"/>                                     2.

<AdministratorID
       Name="Jack"
       Description="Administrator of the HR department computer in HH"
       Profile="ProfileO3"/>                                     2.

<AdministratorID
       Name="Mike"
       Description="Administrator of the QA computer in HH"
       Profile="ProfileO4"/>                                     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="MCHSRVO1"                                       4.
           Description="Remote administration server"
           Address="ftadm://MCHSRVO1.mch.mycompany.net"
           Admission="mchsrvO1remote"/>

        <Instance
           Name="OPENFTO1"                                       4.
           Description="Windows XP"
           Address="ftadm://OPENFTO1.mch.mycompany.net"
           Admission="openftO1remote"/>
```

```
    <Instance
       Name="OPENFT02"                                          4.
       Description="Windows XP"
       Address="ftadm://OPENFT02.mch.mycompany.net"
       Admission="openft02remote"/>

    <Instance
       Name="OPENFT03"                                          4.
       Description="Windows XP"
       Address="ftadm://OPENFT03.mch.mycompany.net"
       Admission="openft03remote">
       <AccessList>                                             5.
          <AccessEntry
             AdministratorID="John"
             DenyFunction="FTMOD"/>
       </AccessList>
    </Instance>

    <Instance
       Name="MCHSRV02"                                          4.
       Description="SUSE Linux 8.1, source management"
       Address="ftadm://MCHSRV02.mch.mycompany.net"
       Admission="mchsrv02remote">
       <AccessList                                              5.
          InheritFromParent="No">
          <AccessEntry
             AdministratorID="Jack"
             AllowFunction="FT FTAC"/>
       </AccessList>
    </Instance>

</Group>

<Group
    Name="MCH2"
    Description="Computer Center Muenchen Freimann">
    <AccessList                                                 5.
       <AccessEntry
          AdministratorID="John"
          AllowFunction="FT FTAC"/>
    </AccessList>
    <Instance
       Name="MCHSRV03"                                          4.
       Description="Windows Server 2003 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"
           Admission="hhwsrv01remote"/>

        <Instance
           Name="HHWSRV02"                                        4.
           Description="HP-11"
           Address="ftadm://HHWSRV02.hhw.mycompany.net"
           Admission="hhwsrv02remote"/>

        <Instance
           Name="HHWSRV11"                                        4.
           Description="Solaris 9"
           Address="HHWSRV11.hhw.mycompany.net"
           Admission="hhwsrv11remote"
           Mode="Legacy">                                         6.
           <AccessList>                                           5.
              <AccessEntry
                 AdministratorID="Mike"
                 DenyFunction="FTAC"/>
           </AccessList>
        </Instance>

    </Group>
```

```
        <Group
            Name="HH2"
            Description="HR department">

            <AccessList>                                          3.
               <AccessEntry
                  AdministratorID="Jack"
                  AllowFunction="FT FTAC"/>
               <AccessEntry
                  AdministratorID="Mike"
                  AllowFunction="FTOP"/>
            </AccessList>

            <Instance
               Name="HHWSRV99"                                    4.
               Description="Mainframe system (BS2000/OSD)"
               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*).

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 openFT commands for the administrator

This chapter contains the commands which are available only to the administrator or which include more options for the administrator than the user or which are primarily used by the administrator.

The commands for the openFT script interface are described in the User Guide as well as in the "openFT Script Interface" manual.

6.1 Overview of the commands

The following overview shows a list of all commands arranged according to the various tasks.

Commands indicated by [b] are primarily aimed at FT users and are therefore only described in the User Guide.

A graphics-capable terminal is required for commands marked [g].

Administer openFT

ftstart	Start asynchronous openFT server
ftstop	Stop asynchronous openFT server
ftshwo	Display operating parameters
ftmodo	Modify operating parameters
ftlang	Change default language setting
install.ftam	Install/deinstall openFT-FTAM
install.ftp	Install/deinstall openFT-FTP
ftsetjava	Manage link to the Java executable
ftshwd	Display diagnostic information
fttrace	Evaluate trace files This command is not described in this chapter but in section “Evaluating trace files with fttrace” on page 380.

Administer partners

ftaddptn	Enter a partner in the partner list
ftshwptn	Display partner properties
ftmodptn	Modify partner properties
ftremptn	Remove a partner from the partner list

Administer key pair sets for authentication

ftcrek	Create key pair set
ftimpk	Import keys
ftshwk	Show key properties

ftmodk	Modify keys
ftupdk	Update public keys
ftdelk	Delete key pair set

Remote administration and ADM traps

ftadm	Enter a remote administration command
ftshwc	Display remote administrable openFT instances
ftshwatp	Display ADM traps
ftexpc	Export configuration of the remote administration server
ftimpc	Import configuration of the remote administration server

File transfer and request queue managing

ncopy [b] / ftscopy	Issue synchronous file transfer request
ft [b] / ftacopy	Issue asynchronous file transfer request
ftcanr	Cancel asynchronous file transfer requests
ftalarm	Report failed requests
ftmodr	Change the order of the requests in the request queue
ftshwr	Display the properties and statuses of requests

Remote command execution

ftexec [b]	Execute operating system commands in remote system

File management

ftcredir [b]	Create remote directories
ftshw [b]	Display attributes of a file / a rdirectory in the remote system
ftshwf [b]	Display the FTAM attributes of a local file
ftmod [b]	Modify file attributes in a remote system
ftmoddir [b]	Modify the attributes of remote directories
ftmodf [b]	Modify the FTAM attributes of a local file
ftdel [b]	Delete a file in a remote system
ftdeldir [b]	Delete remote directories

Logging

ftshwl	Display log records or log files
ftdell	Delete log records or log files
fthelp	Display information on the reason codes in the log records

FTAC function

ftcrep	Create FT profile
ftshwp	Display FT profile
ftmodp	Modify FT profile
ftdelp	Delete FT profile
ftshwa	Display admission set
ftmoda	Modify admission set
ftexpe	Export FT profiles and admission sets
ftshwe	Display FT profiles and admission set from a file
ftimpe	Import FT profiles and admission sets

Administer instances

ftseti [b]	Set an instance
ftshwi [b]	Output information on instances
ftmodi	Modify an instance
ftupdi	Update the instance directory
ftdeli	Deactivate an instance

Display measurement data

ftshwm	Display measurement data of the openFT operation
ftmonitor [g]	Display measurement data of the openFT operation on openFT Monitor

Output of general information and miscellaneous commands

ftinfo [b]	Output information about the openFT system
ftedit[b]	Load local or remote files in the openFT editor

ftmsg[b]	Output message box on a graphical display
openFT	Start openFT Explorer

[b] Command is only described in the User Guide
[g] A graphics-capable terminal is required for this command

As the **administrator**, you may execute the commands listed below with the additional options to perform the corresponding action **system-wide**. This means that:

You can use *ftcanr* to delete any desired file transfer requests.

You can use *ftcrep* to create FT profiles for any login names

You can use *ftdelp* to delete any FT profiles.

You can use *ftmoda* to modify and privilege any of the admission sets.

You can use *ftmodp* to modify any of the FT profiles.

You can use *ftmodr* to change the order of all requests in the request queue independent of the login name.

You can use *ftshwa* to display any of the admission sets.

You can use *ftshwl* to display any of the log records.

You can use *ftshwp* to display any of the FT profiles.

You can use *ftshwr* to obtain information about all the requests for all user IDs.

6.2 Notational conventions

The command syntax essentially corresponds to the output that you get when you specify the command with *-h* option. The following conventions have been used for syntax diagrams:

< > angle brackets are used for parameters which you may replace with current values. You must not specify the angle brackets < > and the permissible value ranges.

[ ] enclose optional entries. The effect on the function of the command is described for the individual parameters.

␣ stands for at least one blank that must be inserted between the various entries.

| stands for alternatives. You may specify only one of the values indicated.

Bold typeface
This is used in the "Description" sections for individual characters or strings that must be specified in exactly the form given, e.g. options or values.
In running text, these are then shown in *italics*.

Lengths and characters sets

The values which you use for parameters in the commands must observe certain restrictions on length and on the characters available:

file name
you can specify an absolute or relative file name.
The file name specified in the local and remote systems may have a maximum length of 512 characters based on the length of the absolute path name. Please note that although long file names can be specified at the openFT interfaces, not all platforms support this maximum length. For example Unix systems permit up to 512 characters whereas Windows systems only permit 256 characters.

If the file name contains blanks, they must be set in double quotes ("), e.g. "file name".

date
numeric; exactly 8 characters in the form `yyyymmdd` with:
`yyyy` for year, `mm` for month and `dd` for day

Note that for all date entries, you may only specify values up to and including 20380119 (January 19, 2038)

user ID

User ID for accessing the required system, maximum 64 characters + 3 characters for hexadecimal format (X' '). The maximum length is system-dependent:
In Unix systems, a maximum of 32 characters with first 8 characters being unique; in Windows systems, a maximum of 36 characters.

command

up to 1000 characters (exception: *ftadm*); for follow-up processing commands, the commands for success and failure must not be longer than 1000 characters in total.

– openFT administers commands using the character set UTF-8 in Windows systems. The maximum lengths for preprocessing, postprocessing or follow-up processing commands (1000 characters) are therefore based on the UTF-8 representation of the corresponding command. In Unix systems, the number of bytes corresponds to the number of characters. In Windows systems, however, the number of bytes may be different from the number of characters because characters that are habitually used but that are not present in the ISO646 character sets (ASCII characters) have a length of two or three bytes in UTF-8 (e.g. the Euro symbol).

partner

Name of the partner system in the partner list (1 to 8 characters) or address of the partner system (maximum 200 characters). The address of the partner system is to be specified in the following form:

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

For further details see section “Specifying partner addresses” on page 68.

profile name

alphanumeric (a..z, A..Z, 0..9), up to 8 characters.

transfer admission

the transfer admission usually consists of printing characters and may not start with a hyphen, minimum 8 characters, maximum 67 characters (in Unix systems, maximum 32 characters). If a transfer admission consists of non-printing characters then it must be specified in hexadecimal format in the form x\'...\' or X\'...\' .

Special characters

Special characters in the entries for *file name, file name-prefix*, *transfer admission*, *user ID*, *account*, *password*, *follow-up processing* (see notes on the commands) must be escaped using a backslash (\). Here, you must differentiate between special characters for file transfer and special characters on a Unix based operating system, and escape the special characters accordingly.

Note that the entries for command strings, file names and free text must be enclosed in single quotes (') or double quotes (").

If the entry for follow-up processing also contains single quotes ('), it is recommended to enclose the entire entry in double quotes ("). The single quotes in the follow-up processing command (e.g. single quotes in a BS2000 password) can then be written as expected in the partner system (such as BS2000).

Example

The account number 1111111,00000000,88888888 is specified in the transfer admission. The comma is a special character that enables file transfer separating the elements of the triple *user ID*, *account* and *password*, and must therefore be escaped with a backslash (\). This backslash is also a special character for the shell, and must therefore also be escaped. The entry then appears as follows:

"1111111\\,00000000\\,88888888"

Sequence of entries

The **sequence** of entries in the command is arbitrary.
Exceptions to this are specifications that do **not** start with a minus sign in the command syntax description if there is more than one such specification (e.g. transfer admission or the system login).

Continuation lines

When there is a large number of parameters, openFT commands can be very long. If you want to use the keyboard to enter commands that are longer than 256 characters, you will need to work with continuation lines. You can obtain these by entering the sequence "\" (backslash) followed by Return.

6.3 Output in CSV format

For some Show commands, openFT offers output in CSV format. CSV (**C**haracter **S**eparated **V**alues) is a popular format in the PC environment in which tabular data is defined by lines. Output in CSV format is offered for the following commands:

– ftshw
– ftshwa
– ftshwatp
– ftshwc
– ftshwe
– ftshwk
– ftshwl
– fshwm
– ftshwo
– ftshwp
– ftshwptn
– ftshwr

Output in CSV format is also possible for the openFT-Script commands *ftshwact* and *ftshws*, see "openFT-Script Interface" manual.

Many programs such as spreadsheets, databases, etc., can import data in CSV format. This means that you can use the processing and presentation features of such programs on the data output by the above commands.

The output fields are described in the appendix starting on page 385.

Every record is output as a line, and each record contains information on an object. If data is present, the first line always contains the header with the field names of each of the columns. **Only the field names are guaranteed, not the order of fields in a record**. In other words, the order of fields is determined by the order of the field names in the header line. Fields within an output line are separated by semicolons (;).

The following data types are differentiated in the output:

Number
: Integer

String
: Since the ";" (semicolon) character has a special meaning in the CSV output as a field separator, a text containing a ";" is enclosed within double quotes. This also applies to the other special characters such as the newline character.
Keywords are never enclosed within double quotes and **always** begin with the character "*" (asterisk).

Date

Date and time are always output in the format `yyyy-mm-dd hh:mm:ss`; a date alone is output in the format `yyyy-mm-dd`.

One example of a possible evaluation procedure is supplied as a reference template in the Microsoft Excel format in the file */opt/openFT/samples/ftaccnt.xlt* . The template evaluates a CSV log file by means of an automatically running macro. The result shows the number of inbound and outbound requests and the Kilobytes transferred in each case for all users.

6.4 ftaddptn - Enter a partner in the partner list

You use the *ftaddptn* command to enter a partner system in the local system's partner list.

Format

ftaddptn -h |
[<partner name 1..8>]
-pa=<partner address 1..200>
[-id=<identification 1..64> | -id=]
[-ri=<routing info 1..8> | -ri=@i | -ri=]
[-ptc=i | -ptc=a | -ptc=]
[-sl=1..100 | -sl=p | -sl=]
[-pri=l | -pri=n | -pri=h]
[-ist=a | -ist=d]
[-am=y | -am=n]
[-rqp=p | -rqp=s]
[-tr=n | -tr=f | -tr=]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

partner name
This is the name to be used to enter the partner system in the partner list. The name may consist of 1 to 8 alphanumerical characters. The first character must be a letter and no distinction is made between uppercase and lowercase. The name can be chosen freely and need only be unique within openFT.

partner name not specified
Specifies that the partner is a dynamic partner.

-pa=partner address
You use *-pa* to enter the address of the partner system in the following form:

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

host (= computer name) is mandatory; all other specifications are optional.

For details concerning address specifications, see section “Specifying partner addresses” on page 68.

-id=identification | **-id=**
Identification unique in the network of the openFT instance in the partner system. In the case of FTAM partners, it is possible to specify an Application Entity Title in the form n1.n2.n3.n4..mmm as the identification. n1, n2 etc. are positive integer values which describe the "Application Process Title". n1 can only have the values

0, 1 or 2, n2 is restricted to values between 0 and 39 if n1 does not have the value 2. The optional Application Entity Qualifier mmm must be separated from the values of the Application Process Title by two periods. For details, see the openFT User Guide.

-id must not be specified for FTP partners!

Identification not specified
The specification of *-id=* means that the *host* (host name) is used for identification for the openFT and FTADM protocol.

Default value: *host* (host name) for the openFT and FTADM protocol, otherwise blank.

-ri=routing info | **-ri=@i** | **-ri=**
If the partner system can only be accessed via an intermediate instance then you specify the address information to be used for routing by the intermediate instance in *routing info*.

@i for *routing info*
The instance identification specified in *-id=* is used as the routing information.

neither *@i* nor *routing info* specified (default value)
The specification of *-ri=* (without parameters) means that the partner system can be accessed directly, i.e. without an intermediate instance.

-ptc=i | **-ptc=a**
You can use *-ptc* to modify the operating parameter setting for sender verification on a partner-specific basis. These settings only affect partners which are connected via the openFT protocol and do not operate with authentication (e.g. partners with openFT V8.0 or earlier).

i (identification)
Deactivates checking of the transport address. Only the partner's identification is checked. The partner's transport address is also not checked even if extended sender verification is globally active (see the *ftmodo* command on page 229).

a (address)
Activates checking of the transport address. The partner's transport address is checked even if checking of the transport address is globally deactivated (see *ftmodo* command on page 229).

If the transport address under which the partner logs on is not the same as the entry in the partner list then the request is rejected.

neither *i* nor *a* specified (default value)
-ptc= (without parameters) means that the operating system parameters apply to sender verification.

-sl=1..100 | **-sl=p** | **-sl=**
You use this option to assign a security level to the partner system.

A low security level means that the need for protection vis a vis this partner is low, for instance because the partner's identity has been authenticated using cryptographic methods, which means that you can be certain that the partner is genuinely who they claim to be.

A high security level means that the need for protection vis a vis this partner is high, because the identity of the partner has only been determined on the basis of their address, for instance, and that no authentication has been performed using cryptographic methods.

1..100
Assigns a fixed security level to the partner. 1 is the lowest and 100 the highest security level.

All integers 1 through 100 are permitted.

p Assigns a security level to the partner on the basis of the partner's attributes, i.e.:

- Security level 10 if the partner has been authenticated.
- Security level 90 if the partner is known in the transport system and is identified by the name it is known by in the transport system.
- Security level 100 if the partner has only been identified by its address.

Security level not specified (default value)
-sl= (without parameters) means that the operating parameter setting for the security level applies (see command *ftmodo* on page 229).

-pri=l | **-pri=n** | **-pri=h**
-pri allows you to specify the priority of a partner in respect of processing requests that have the same request priority. This means that the partner priority only applies in the case of requests that have the same request priority, but that are issued to partners with a different partner priority.

l (low)
The partner is assigned a low priority.

n (normal, default)
The partner is assigned a normal priority.

h (high)
The partner is assigned a high priority.

st=a | **-st=d** | **-st=ad**

This option allows you to control how locally submitted asynchronous file transfer requests to the specified partner system are processed.

a (active, default value)

Locally submitted asynchronous file transfer requests to this partner system are processed if the asynchronous openFT server is started.

d (deactivated)

Locally submitted asynchronous file transfer requests to this partner system are initially not processed but are stored in the request queue.

ad (automatic deactivation)

Multiple consecutive unsuccessful attempts to establish a connection to this partner system result in its deactivation. The maximum number of unsuccessful attempts is 5. If you want to perform file transfer again with this system, you must explicitly activate it with *ftmodptn -st=a*.

The maximum number of such unsuccessful attempts is 5. After a connection has been established successfully, the counter is reset to 0.

-ist=a | **-ist=d**

This option allows you to control how file transfer requests issued remotely by the specified partner system are processed.

a (active, default value)

File transfer requests issued remotely by this partner system are processed if the asynchronous openFT server is started.

d (deactivated)

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

-am=n | **-am=y** You can use this option to force partner authentication.

n (default value)

Authentication is not forced, i.e. this partner is not restricted with regard to authentication.

y Authentication is forced, i.e. requests are only processed if the local system is successfully able to authenticate the partner, see page 77.

-rqp=p | **-rqp=s**

You use this option (rqp = request processing) to control whether asynchronous outbound requests to this partner are always run serially or whether parallel requests are permitted.

p (parallel, default value)
Parallel connections to this partner are permitted.

s (serial)
Parallel connections to this partner are not permitted. If multiple file transfer requests to this partner are pending then they are processed serially. A follow-up request is not started until the preceding request has terminated.

-tr=n | **-tr=f** | **-tr=**
You can use this option to modify the operating parameter settings for the partner selection for the openFT trace function on a partner-specific basis.

n (on)
The trace function is active for this partner. However, a trace is only written if the openFT trace function has been activated via the operating parameters. In this case, this setting for *ftaddptn* takes priority over the partner selection for the trace function in the operating parameters. See page 229ff, *ftmodo*, *-tr* option.

f (off)
The trace function is deactivated for this partner.

neither *n* nor *f* specified (default value)
-tr= (without parameters) means that the global setting for partner selection in the openFT trace function applies (see *ftmodo* command on page 229).

6.5 ftadm - Execute remote administration command

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

In addition, as remote administrator, you can use *ftadm* to query the following information from the remote administration server (see the section “Remote administration commands” on page 179):

– You can determine what openFT instances you are authorized to administer and what remote administration permissions you have for these instances.

– You can read the ADM traps that the openFT instances you are administering have sent to the remote administration server. For this to be possible, the remote administration server must also be configured as an ADM trap server for the administered openFT instances. For details, see the section “ADM traps” on page 147.

Format

```
ftadm  -h |
       [ -c ]
       [ -cs=<partner 1..200> ]
       [ -ri=<routing info 1..200> ]
       <command 1..8192> | -
       [ <transfer admission 8..67> | @d ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-c Specifies whether the user data (i.e. the command and the command output) is to be transferred in encrypted form. It is only possible to specify *-c* if openFT-CR is installed. If openFT-CR is not installed, *-c* is suppressed in the command syntax (*-h*) and a syntax error is generated if *-c* is specified.

-cs=partner
Specifies the name of the remote administration server in the partner list or the address of the remote administration server. The remote administration server must be addressed as an ADM partner. For details, see the section “Specifying partner addresses” on page 68.

-cs not specified

If you do not specify *-cs*, it is assumed that the local system, i.e. the system at which you logged on, is the remote administration server. This means that you can only omit *-cs* if you enter *ftadm* directly on the remote administration server.

-ri=routing info

Specifies the pathname of the openFT instance that you want to administer. The pathname is configured by the ADM administrator on the remote administration server and is required in order to forward the remote administration request to the openFT instance. You can get the pathname by running the command *ftshwc* on the remote administration server. See the section “Remote administration commands” on page 179.

-ri not specified

If you do not specify *-ri*, the command specified under *command* is executed on the remote administration server, e.g. *ftshwc* or *ftshwatp*. See section “Remote administration commands” on page 179.

command

The remote administration command to be executed. The maximum command length supported is 8192 characters.

- (dash) for *command*

The dash stands for the standard input *stdin*, i.e. you enter the command at the keyboard. Terminate your input by pressing <END> or CTRL+D.

If input is blanked (*@d*) for the *transfer admission*, the system first queries the transfer admission. You can then enter the command.

transfer admission | **@d**

FTAC transfer admission for accessing the remote administration server. Specification of the transfer admission is mandatory if you have specified *-cs*, and must not be specified if you have not specified *-cs*.

@d for *transfer admission*

If you specify *@d* (blanked), the transfer admission is queried on screen after the command has been sent. The entry you make is not displayed, in order to prevent unauthorized persons from seeing the transfer admission.

transfer admission not specified

If you do not specify an FTAC transfer admission, two possible situations arise:

– If you have also specified *-cs*, the transfer admission is queried on screen after the *ftadm* command has been sent.
– If you do not specify *-cs*, i.e. if you enter *ftadm* directly at the remote administration server, your user ID is used as proof that you are authorized to perform remote administration.

6.5.1 Remote administration commands

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

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

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

In the case of a remote administration request, these permissions are compared with the permissions you have on the relevant instance as a remote administrator. The ADM administrator defines the permissions in the configuration data of the remote administration server.

If your permissions are insufficient to execute the remote administration command on a particular instance, the request is rejected, e.g. with:

```
ftadm: Administration request rejected by remote administration server
```

In this event, an ADM log record is written on the remote administration server with a reason code not equal to 0000. The reason code specifies the exact reason for rejection (*fthelp reason-code*).

Commands for openFT partners in BS2000/OSD

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

BS2000 command	Short forms and aliases	Permission
ADD-FT-PARTNER	ADD-FT-PART FTADDPTN	FT
CANCEL-FILE-TRANSFER	CAN-FILE-T, CNFT NCANCEL, NCAN FTCANREQ	FT
CREATE-FT-KEY-SET	CRE-FT-KEY FTCREKEY	FT
CREATE-FT-PROFILE	CRE-FT-PROF	FTAC
DELETE-FT-KEY-SET	DEL-FT-KEY FTDELKEY	FT
DELETE-FT-LOGGING-RECORDS	DEL-FT-LOG-REC FTDELLOG	FT | FTAC
DELETE-FT-PROFILE	DEL-FT-PROF	FTAC
IMPORT-FT-KEY [2)]	IMP-FT-KEY FTIMPKEY	FT
MODIFY-FILE-TRANSFER	MOD-FILE-T FTMODREQ	FT
MODIFY-FT-ADMISSION-SET	MOD-FT-ADM	FTAC
MODIFY-FT-KEY [2)]	MOD-FT-KEY FTMODKEY	FT
MODIFY-FT-OPTIONS	MOD-FT-OPT FTMODOPT	FT
MODIFY-FT-PARTNER	MOD-FT-PART FTMODPTN	FT
MODIFY-FT-PROFILE	MOD-FT-PROF	FTAC
REMOVE-FT-PARTNER	REM-FT-PART FTREMPTN	FT
SHOW-FILE-TRANSFER	SHOW-FILE-T, SHFT NSTATUS, NSTAT FTSHWREQ	FT | FTOP
SHOW-FT-ADMISSION-SET	SHOW-FT-ADM-S	FTAC
SHOW-FT-DIAGNOSTIC	SHOW-FT-DIAG FTSHWD	FT | FTOP | FTAC
SHOW-FT-INSTANCE	SHOW-FT-INST	FT | FTOP

BS2000 command	Short forms and aliases	Permission
SHOW-FT-KEY [2)]	FTSHWKEY	FT | FTOP
SHOW-FT-LOGGING-RECORDS	SHOW-FT-LOG-REC FTSHWLOG	FT | FTOP | FTAC
SHOW-FT-MONITOR-VALUES [1)]	SHOW-FT-MON-VAL FTSHWMON	FT | FTOP
SHOW-FT-OPTIONS	SHOW-FT-OPT FTSHWOPT	FT | FTOP
SHOW-FT-PARTNERS	SHOW-FT-PART FTSHWPTN	FT | FTOP
SHOW-FT-PROFILE	SHOW-FT-PROF	FTAC
START-FTTRACE	FTTRACE	FT | FTOP
STOP-FT	FTSTOP	FT
UPDATE-FT-PUBLIC-KEYS	UPD-FT-PUB-KEY FTUPDKEY	FT

[1)] As of V11.0

[2)] As of V12.0

Commands for openFT partners in z/OS

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

[1)] As of V11.0

[2)] As of V12.0

Commands for openFT partners in Unix and Windows systems

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

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

Commands on the remote administration server

ftadm allows you to execute the commands *ftshwc* and *ftshwatp* on the remote administration server. When you do so, you must not specify the *-ri* option:

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

6.6 ftalarm - Report failed requests

The *ftalarm* command is used to trigger an alarm if, within two minutes, more FT requests than the number specified by the user fail. The failed FT requests are identified using a return code not equal to 0 for the FTAC log records. *ftalarm* uses the *cron* function.

A separate *ftalarm* call is required for each instance.
Proceed as follows: activate the instance with *ftseti*, and call *ftalarm*.

If *ftalarm* is started on Solaris systems via SMF then it is inadvisable to start the ftalarm command manually since SMF is not informed of any changes. For SMF, *ftalarm* is a so-called transient service, i.e. there is no process that can be monitored.

Format

ftalarm [-h |
-s <number of errors 1..99999999> |
-t]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-s number of errors
starts the *ftalarm* function. When the specified *number of errors* in FTAC log records is exceeded within two minutes, the following message is output on the console and to the *conslog* file:

```
openFTalarm: number or more access control error loggings within 2
minutes
```

The partial string *openFTalarm:* within this message is also guaranteed for future versions of openFT and can be interpreted for automatic processing by system management tools.

The messages are output by the *cron* function at regular intervals and can therefore be delayed by up to one minute when the *ftalarm* function is activated.

conslog is located in the *log* directory of the relevant openFT instance. In the case of the default instance, the pathname is */var/openFT/std/log/conslog*.

-t terminates the *ftalarm* function.

6.7 ftcanr - Cancel asynchronous requests

You can use the *ftcanr* command to cancel asynchronous requests which are in the course of being processed or which are waiting to be processed in the request queue. As an ordinary FT user, you can only cancel requests entered under your own login name.

The FT administrator can cancel any requests. In addition, as administrator you can delete requests unconditionally, i.e. without negotiating with the partner system.

If file transfer requests have already been started, the status of the destination file may be undefined.

Format

```
ftcanr  -h  |
        [ -f ]
        [ -ua=<user ID 1..32> | @a ]
        [ -ini=l | -ini=r | -ini=lr | -ini=rl ]
        [ -pn=<partner 1..200> ]
        [ -fn=<file name 1..512> ]
          <request ID 1..2147483647> [<request ID 1..2147483647> ...] | @a
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-f *-f* allows you to delete the request unconditionally from the local request queue, i.e. without negotiation with the partner system. Note that this can cause requests with an undefined state to arise in the partner's request queue.

You can only call this option as FT administrator. The precondition is that the request was first cancelled with *ftcanr* without the option *-f*.

-ua=user ID | **@a**

You use *-ua* to indicates the user ID for which requests are to be cancelled.

user ID

The FT administrator can specify any login name.

@a This option is only significant for the FT administrator. The FT administrator can specify *@a* to cancel the requests of all the login names.

-ua= not specified

Your login name is used as the selection criterion. Exception: The FT administrator has called the command and specified transfer IDs. In this case, the default is *@a*.

-ini=l | **-ini=r** | **-ini=lr** | **-ini=rl**

You use *-ini* to indicate the initiating party for which you want to cancel requests. You can specify *l*, *r*, *lr*, *rl*:

l Only requests initiated locally are cancelled.

r Only requests initiated remotely are cancelled.

lr, rl Both local and remote requests are cancelled.

-ini not specified

The initiator is not used as a selection criterion (corresponds to *lr* or *rl*).

-pn=partner

You use *-pn* to specify the partner system for which you want to cancel requests. *Partner* is the name or address of the partner system. You should specify the partner in the same form as in the request allocation or as in the output from the *ftshwr* command.

-fn=file name

You use *-fn* to specify the name of the file for which requests are to be cancelled. Requests which access this file in the local system are cancelled.
You must specify the file name which was used when the request was issued and which is output for the *ftshwr* command. Wildcards are not permitted in file names.

request ID1 [request ID2] [request ID3] ... | **@a**

For *request ID*, enter the number of the request to be cancelled. Leading zeros may be omitted. The request identification *request ID* may be obtained from the request receipt acknowledgment displayed on the screen, or using the *ftshwr* command if you have forgotten the *request ID*. You can also specify a number of request identifications at the same time.
If, in addition to *request ID*, you specify other selection criteria, a request with the specified *request ID* is only cancelled if it also satisfies the other conditions.

@a specified as *request ID*

@a selects all requests.

If request IDs were specified and the other selection criteria specified are not satisfied by the requests, the request is not cancelled and the following new error message is issued:

`ftcanr: Request` *request ID* `not found`

request ID is the identification of the last unsuitable request.

Examples

1. The asynchronous request with request identification 65546 should be deleted.

   ```
   ftcanr␣65546
   ```

2. All local requests to the partner *ux1* which relate to the file *file1* should be deleted.

   ```
   ftcanr -pn=ux1 -fn=file1 -ini=l @a
   ```

6.8 ftcrei - Create or activate an instance

The *ftcrei* command allows you to create a new instance or re-activate a deactivated instance.

When an instance is created, the instance file tree is linked to the */var/openFT* directory with the resources of an instance.

If the specified instance file tree does not yet exist, it is created.

When the instance file tree is created, the operating parameters, the profile files and the startup and shutdown files (not in the case of Solaris with SMF) are initialized in the same way as for a new installation. In the case of Solaris with SMF, a manifest is generated and entered in SMF, see section “Solaris SMF” on page 41.

If the instance file tree already exists, *ftcrei* checks the version. If the instance file tree was created using an older version of openFT, it must first be updated using the *ftupdi* command before it can be reactivated.

Important notes for when using multiple instances

– Use of several openFT instances is only possible using the TCP/IP transport system. If you would like to use several instances and are working with CMX with TNS activated (*ftmodo -cmx=y -tns=y*), you must delete all openFT-specific TNS entries that are not TCP/IP compliant (i.e. all except for LANINET and RFC1006).

– You must explicitly assign an individual address to all instances using *-addr*.

– If the instance is to be authenticated in partner systems, it must have a unique instance ID assigned to it (using *fta -id=*). In addition, a public key for the instance must be made available to the partner systems.

Format

ftcrei -h |
<instance 1..8> [<directory 1..128>][-addr=<host name>]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

instance
Name of the instance to be created.
Instance names have a maximum length of 8 characters. The permitted characters are A-Z, a-z and 0-9, and the first character must not be numerical.

The instance name must not be confused with the instance ID (see *ftmodo -id=*).

directory
: Directory in which the instance file tree is to be located. The directory must not yet exist.

  If you do not specify *directory*, the instance file tree is by default created in:

  `/var/openFT/.`*instance*

-addr=host name
: Internet host name by which the instance is addressed. If your system has a DNS name, you should specify the full DNS name. openFT then uses the first 8 characters of the first part of the name (the host name qualifier) as the processor name (*ftmodo -p=*) and the entire name as the instance ID (*ftmodo -id=*).

Messages of the ftcrei command

If *ftcrei* could not be executed properly, a self-explaining message is output. The exit code is not equal zero in this case.

Examples

1. The instance *inst1* is to be newly created in the directory */cluster/inst1*. The DNS name is *hugo.abc.net*. The directory */cluster/inst1* is not allowed to exist.

   ```
   ftcrei inst1 /cluster1/inst1 –addr=hugo.abc.net
   ```

   Where the operational parameter *ftmodo -p=* is *hugo* and *ftmodo -id=* is *hugo.abc.net*.

2. The existing instance *inst2* from the directory */cluster/inst2* is to be re-activated. No host name may be specified.

   ```
   ftcrei inst2 /cluster/inst2
   ```

6.9 ftcrek - Create key pair set

You use this command to create a key pair set for the authentication of your openFT instance in partner systems (RSA procedure). For more information on administering keys, see the section “Authentication” on page 77.

If the maximum number of key pair sets is exceeded you get the error message:

```
ftcrek: Maximum number of key pairs exceeded
```

Format

ftcrek [-h]

Description

-h Displays the command syntax.

6.10 ftcrep - Create an FT profile

ftcrep stands for "create profile". This command can be used by any user to set up FT profiles for his or her login name.

The FTAC administrator can also set up FT profiles for other login names, either with or without defining a transfer admission.

When it is created, the profile is given a timestamp that is updated each time the profile is modified (e.g. using *ftmodp*).

Format

```
ftcrep  -h |
          <profile name 1..8> | @s
          <transfer admission 8..32> | @n
        [ -ua=[<user ID 1..32>] [,[<password 1..20> | @n ]] ]
        [ -v=y | -v=n ] [ -d=yyyymmdd ]
        [ -u=pr | -u=pu ]
        [ -priv=y | -priv=n ]
        [ -iml=y | -iml=n ]
        [ -iis=y | -iis=n ] [ -iir=y | -iir=n ]
        [ -iip=y | -iip=n ] [ -iif=y | -iif=n ]
        [ -ff=[t][m][p][r][a][l] | -ff=c ]
        [ -dir=f | -dir=t | -dir=ft ]
        [ -pn=<partner 1..200>,...,<partner(50) 1..200> | -pn= ]
        [ -fn=<file name 1..512> | -fn= ]
        [ -fnp=<file name prefix 1..511> ]
        [ -ls= | -ls=@n | -ls=<command1 1..1000> ]
        [ -lsp=<command2 1..999> ] [ -lss=<command3 1..999> ]
        [ -lf= | -lf=@n | -lf=<command4 1..1000> ]
        [ -lfp=<command5 1..999> ] [ -lfs=<command6 1..999> ]
        [ -wm=o | -wm=n | -wm=e | -wm=one ]
        [ -c=y | -c=n ]
        [ -txt=<text 1..100> ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

profile name | **@s**
: is the name you wish to assign to the FT profile. This name can be used to address the FT profile, for example when it is to be modified or deleted. Be sure not to confuse the profile name with the transfer admission (see below). The profile name must be unique among all the FT profiles under your login name, or FTAC will reject the *ftcrep* command and issue the message *FT profile already exists*. To have the profile names you have already assigned displayed, you can issue the *ftshwp* command (without options).

  @s for *profile name*
  : Creates the standard admission profile for the user ID. You must specify *@n* as the transfer admission, because a standard admission profile in a request is addressed using the user ID and password.

    You must not specify the options *-v*, *-d* and *-u* with a standard admission profile.

transfer admission | **@n**
: replaces the login authorization for your Unix system otherwise required in inbound requests. When this transfer admission is specified in an FT request, FTAC applies the access rights defined in this FT profile.

  transfer admission
  : The transfer admission must be unique within your Unix system so that there are no conflicts with transfer admissions defined by other FTAC users with other access rights. If the transfer admission you select has already been assigned, FTAC rejects the *ftcrep* command and issues the message: `Transfer admission already exists.`

    You can also define a binary admission with any characters, including non-printing characters. To do this, you must specify the transfer admission in hexadecimal format in the following form: x\'...\' or X\'...\', e.g. x\'f1f2f3f4f5f6f6f8\'.

    As the FTAC administrator, you can assign a transfer admission for yourself under your own login name or for any other user.
    In this case, if you do not have FT administrator permissions, you must specify the complete login admission, i.e. the user ID and password.

  @n for *transfer admission*
  : By entering *@n*, you create an FT profile without a transfer admission.

    As the FTAC administrator, by specifying *@n*, you can create FT profiles for other login names without having to define transfer admissions.

If the profile is not a standard admission profile, it is locked until you or the owner of the profile assign a valid transfer admission with *ftmodp*.

You must specify *@n* when you create a standard admission profile.

transfer admission not specified
: If you do not specify the transfer admission in the command, FTAC prompts you to enter the transfer admission after the command has been sent. Your entry is not displayed to prevent unauthorized persons from seeing the transfer admission. To exclude the possibility of typing errors, the program expects you to enter the transfer admission a second time as an entry check.

-ua=[user ID][**,**[password | **@n**]]
: As the FTAC administrator use *-ua* to specify the user IDs for which you want to set up FT profiles.

  user ID
  : The user without administrator privileges can specify only his own user ID. As the FTAC administrator, you can specify any user ID.

  ,password
  : Specifies the password of the login name. A binary password must be specified in hexadecimal format in the form x\'...\' or X\'...\'. The FT profile for the login name is only valid while the password is valid for the login name. If the password is changed, the profile can no longer be used.

    If you want to assign an FT profile for another user and also assign a transfer admission for that profile, you must specify the login name as well as the password for that login name if you do not have FT administrator privileges.

  @n for *password*
  : This entry may only be specified by the FTAC administrator. With *@n*, you cannot assign any transfer admission for the FT profile if you do not have FT administrator privileges.

  comma only (,) no *password*
  : Entering comma (,) without *password* causes FTAC to query the password on the screen after the command is entered. The entry is not displayed to prevent unauthorized persons from seeing the transfer admission. In this case, quotes must not be escaped with a backslash (\).

  user ID only (without comma and no *password*) specified
  : the profile is valid for all the passwords for *user ID*.

-ua=␣ specified or *-ua* not specified
: the FT profile is created for the individual login name.

-v=y | **-v=n**
defines the status of the transfer admission.

Possible values are:

y (default value)
the transfer admission is not disabled (it is valid).

n the transfer admission is disabled (it not valid).

-v must not be specified with a standard admission profile.

-d=yyyymmdd
specifies the period during which the transfer admission can be used. The FT profile is disabled when this period has expired.
You can specify an eight-digit date (e.g. 20170602 for June 2, 2017). The transfer admission can no longer be used after 00:00 hours on the specified day. The largest possible value which can be specified as the date is 20380119 (January 19, 2038).

-d must not be specified with a standard admission profile.

-d not specified (default value)
no period is specified for using the transfer admission.

-u=pr | **-u=pu**
with *-u*, you can control how FTAC reacts when someone attempts to create an FT profile with the same transfer admission. Normally, the transfer admission must be disabled immediately.

Transfer admissions that do not require as much protection are designated as public. This means that they are not disabled, even if a user attempts to assign another transfer admission of the same name.

pr (default value)
the transfer admission is disabled as soon as someone under another login name attempts to specify a transfer admission of the same name (private). In this case, the values for *-u* and *-d* are set to their default values at the same time.

pu the transfer admission is not disabled, even if someone attempts to specify a transfer admission of the same name (public).

-u must not be specified with a standard admission profile.

-u not specified
The previous setting remains unchanged.

-priv=n | **-priv=y**
is used by the FTAC administrator to grant privileged status to FT profiles.

n (default value)
The FT profile is not privileged (initially).

y The FT profile is privileged.

-iml=y | **-iml=n**
-iml (ignore max. level) is used to specify whether the FT profile is to be restricted by the values in the admission set. You can override your own the entries (the MAX. USER LEVELS) for requests using this FT profile.

If the FT profile is also privileged by the FTAC administrator, the values of the FTAC administrator (the MAX. ADM LEVELS) can also be ignored. This FT profile would then allow *inbound* basic functions which are disabled in the admission set to be used. Possible values are:

y allows the values in the admission set to be ignored.

n (default value)
restricts the functionality of the profile to the values in the admission set.

-iis=y | **-iis=n**
-iis (ignore inbound send) allows the value for the basic function *inbound send* in the admission set to be ignored (for details, see *-iml*).

y allows the basic function *inbound send* to be used even if it is disabled in the admission set. At the same time, the component "display file attributes" of the basic function *inbound file management* can also be used.

Specifying this option is enough as long as the basic function *inbound send* was disabled by the user, but if it was disabled by the FTAC administrator, it is also necessary that he/she grant privileged status to the FT profile.

n (default value)
restricts the profile to the value in the admission set for the basic function *inbound send*.

-iir=y | **-iir=n**
-iir (ignore inbound receive) allows the value for the basic function *inbound receive* in the admission set to be ignored (for details, see *-iml*).

y allows the basic function *inbound receive* to be used even if it is disabled in the admission set. At the same time, components of the basic function *inbound file management* can also be used (see table at *-iif*).

Specifying this option is enough as long as the basic function *inbound receive* was disabled by the user, but if it was disabled by the FTAC administrator, it is also necessary that he/she grant privileged status to the FT profile.

n (default value)
restricts the profile to the value in the admission set for the basic function *inbound receive*.

-iip=y | -iip=n
-iip (ignore inbound processing) allows the value for the basic function *inbound follow-up processing + preprocessing + postprocessing* in the admission set to be ignored (for details, see *-iml*).

y allows the basic function *inbound follow-up processing + preprocessing + postprocessing* to be used even if it is disabled in the admission set. Specifying this option is enough as long as the basic function *inbound receive* was disabled by the user, but if it was disabled by the FTAC administrator, it is also necessary that he/she grant privileged status to the FT profile.

n (default value)
restricts the profile to the value in the admission set for the basic function *inbound follow-up processing + preprocessing + postprocessing*.

-iif=y | -iif=n
-iif (ignore inbound file management) allows the values for the basic function *inbound file management* in the admission set to be ignored (for details see *-iml*).

y allows the basic function *inbound file management* to be used even if it is disabled in the admission set. Specifying this option is enough as long as the basic function *inbound file management* was disabled by the user, but if it was disabled by the FTAC administrator, it is also necessary that he/she grant privileged status to the FT profile.

n (default value)
restricts the profile to the value in the admission set for the basic function *inbound file management*.

The following table shows which subcomponents of the file management can be used under which conditions.

Inbound file management function	**Values of the admission set or extension in profile**
Display file attributes	Inbound Send (IBS) enabled
Modify file attributes	Inbound Receive (IBR) **and** Inbound File Management (IBF) enabled
Rename files	Inbound Receive (IBR) **and** Inbound File Management (IBF) enabled
Delete files	Inbound Receive (IBR) enabled **and** Write mode = overwrite in profile
Display directories	Inbound File Management (IBF) enabled

Inbound file management function	**Values of the admission set or extension in profile**
Create, rename and delete directories	**Inbound File Management (IBF) enabled and** direction= from partner in profile

-ff=[t][m][p][r][a][l] | -ff=c

-ff defines the FT function for which the FT profile can be used. With the exception of *c*, these letters can be combined in any way (*tm, mt, mr,* ...). *c* must not be combined with other values.

t (transfer) The FT profile can be used for the file transfer functions "Transfer files", "Display file attributes", and "Delete files".

m (modify file attributes) The FT profile can be used for the file transfer functions "Display file attributes" and "Modify file attributes".

p (processing) The FT profile can be used for the file transfer functions "File Preprocessing" or "File Postprocessing". The FT function "Transfer files" must also be permitted.

Specification of *p* has no significance for profiles with a file name prefix (*-fnp=*) or a file name (*-fn=*) since, in this case, the first character of the file name or file name prefix decides whether the profile can only be used for preprocessing and postprocessing ("|") or only for file transfer/file management (no "|").
The use of follow-up processing is not controlled by *-ff=*, but by *-lf=* and *-ls=*.

r (read directory) The FT profile can be used for the file transfer functions "Display directories" and "Display file attributes".

a (administration)
The admission profile is allowed to be used for the "remote administration" function. This means that it authorizes a remote administration server to access the local openFT instance. To do this, the associated transfer admission must be configured in the remote administration server.
-ff=a may only be specified by the FT administrator or FTAC administrator.

l (logging)
The admission profile is allowed to be used for the "ADM traps" function. This allows another openFT instance to send its ADM traps to the remote administration server via this profile. This specification only makes sense if the local openFT instance is flagged as a remote administration server (*ftmodo -admcs=y* command).
-ff=l may only be specified by the FT administrator.

c (client access)
The admission profile is allowed to be used for the "access to remote administration server" function (ADM profile). This allows a remote administrator on a remote computer to use this profile to access the local remote administration server and issue remote administration requests. The local openFT instance must be flagged as a remote administration server (*ftmodo -admcs=y* command).

The value *c* must not be combined with any other value. *-ff=c* may only be specified by the ADM administrator.

-ff not specified
Corresponds to the specification *-ff=tmr*, i.e. the admission profile can be used for all file transfer functions other than "file processing", but cannot be used for remote administration functions (*a*, *c*) and ADM traps (*l*).

-dir=f | **-dir=t** | **-dir=ft**
specifies for which transfer direction(s) the FT profile may be used.

f allows data transfer only from a remote system to the local system.

t allows data transfer only from a local to a remote system. Directories cannot be created, renamed nor deleted.

ft, tf both transfer directions are allowed.

-dir not specified
transfer direction is not restricted in the FT profile.

-pn=partner[**,**partner2**,** ...] | **-pn=**
You use *-pn* to specify that this admission profile is to be used only for FT requests which are processed by a certain partner system. You can specify the name of the partner system in the partner list or the address of the partner system. For details on address specifications, see section “Specifying partner addresses” on page 68.

You can specify more than one partner system (maximum 50) with a maximum total of 1000 characters.

-pn not specified (or **-pn=**)
means that any remote system can use the FT profile.

-fn=file name | **-fn=**

-fn specifies which file under your login name may be accessed using this FT profile. If you specify a fully qualified file name, only the file with this name can be transferred.
If the file name ends with %unique or %UNIQUE, this string is replaced during the file transfer by a string which changes for each new call. In Unix systems, this string is 14 characters long. In addition, a suffix separated by a dot may be specified after %unique or %UNIQUE, e.g. `file1%unique.txt`. Only the already converted file name is displayed in both the log and the messages.
If *file name* starts with a "|" (pipe character) then it is interpreted as a preprocessing or postprocessing command, see also the corresponding section in the user guide.

-fn not specified (or **-fn=**)

omitting *-fn* means that the FT profile allows unrestricted access to all files under the login name (exception see *-fnp*).

-fnp=file name prefix

restricts access to a set of files whose names begin with the same prefix. FTAC adds the character string specified as *file-name-prefix* to the file name in the request and attempts to transfer the file with the expanded name. For example, if this option is specified as *-fnp=scrooge/* and the request contains the file name *stock*, the file transferred is *scrooge/stock*).

In this way, you can designate the files you have released for transfer. If the *-fnp* option was used to specify a prefix, the file name specified in the request must not contain the character string *../*. This disables (unintentionally) changing directories. You should also ensure that there is no chance for a symbolic link to cause a jump to another place in the file tree.

%unique or %UNIQUE cannot be used for a file name prefix. In the case of a file transfer request, the user can use a file name ending with %UNIQUE (or %UNIQUE.*suffix* or %unique or %unique.*suffix*) to generate a unique file name with the prefix specified here.

A file name prefix which starts with the | (pipe) character indicates that the FTAC profile can only be used for file transfer with preprocessing and postprocessing, since the file name created using the prefix and the name specified for the *ncopy* or *ft* command also starts with the | character. In this case, no follow-up commands may be specified.

On Unix systems, the shell metacharacters | ; & < > and "newline" may only be specified if they are enclosed in '...' (single quotes) or "..." (double quotes) or if each of them is escaped with "\" (backslash). The character ` (accent grave) and the string $((dollar+open bracket) may only be specified if they are enclosed in '...' (single quotes) or if they are specified directly after a backslash ("\").

The following strings may not be specified for the name entered in the *ncopy* or *ft* command:
– .. (two dots)
– .\ (dot + backslash)
– .' (dot + single quote)
This makes it impossible to navigate to higher-level directories.

filename prefix can be up to 511 bytes in length (for the representation in UTF-8, see page 163).

Special cases

- You must specify a file name or file name prefix which starts with the string "|ftexecsv␣" for FTAC profiles which are to be used exclusively for the *ftexec* command.
  If a command prefix is also to be defined, you must specify it as follows:

  -fnp="|ftexecsv␣-p=*command prefix***"**
  (e.g.: `-fnp="|ftexecsv␣-p=\"ftshwr␣\""`)

  The same restrictions apply to the command string of the *ftexec* call as to the filename prefix during preprocessing and postprocessing.

- For FTAC profiles that are only to be used for getting monitoring data, specify the filename prefix "|*FTMONITOR ". The functions of the profile must permit File Preprocessing (*-ff=tp*). For details, see Example 3 on page 201.

-fnp not specified
FTAC adds no prefix to the file name.

-ls= | **-ls=@n** | **-ls=**command1
-ls specifies follow-up processing which is to be performed under your login name in the event that file transfer is successful. If *-ls* is specified, no success follow-up processing may be requested in the FT request. Specifying *-ls* only makes sense if you also make an entry for *-lf* (see below) to preclude the possibility than an intentionally unsuccessful request can circumvent the *-ls* entry. If you have defined a prefix for the file name with *-fnp* and plan follow-up processing for this file, you must specify the complete file name here.

@n for *command1*
If *-ls=@n* is specified, no success follow-up processing is permitted in the event of a successful file transfer.

-ls not specified (or **-ls=**)
does not restrict follow-up processing in the local system in the event of successful file transfer (however, see also *-lsp* or *-lss*).

-lsp=command2

-lsp defines a prefix for follow-up processing in the local system in the event of successful file transfer. FTAC then adds the character string *command2* to the follow-up processing specified in the FT request and attempts to execute the resulting command.
For example, if this option is specified as *-lsp='lpr␣'* and the request specifies *file-name* as follow-up processing, FTAC executes *lpr␣file-name* as follow-up processing.

Prefix and suffix and follow-up processing command must together not be longer than 1000 bytes (for the representation in UTF-8, see page 163).

Please also bear in mind the information provided on the *-ls* option!

If a prefix was defined with *-lsp*, the character set available for specifying follow-up processing in the FT request is restricted to:
– alphanumeric characters (letters and digits)
– the special characters + = / ! ␣ - , @ _ " $ '
– a period (.) between alphanumeric characters

-lsp not specified

FTAC adds no prefix to the follow-up processing specified in the request in the event of successful file transfer.

-lss=command3

-lss defines a suffix for follow-up processing in the local system in the event of successful file transfer. FTAC then appends the character string *command3* to the follow-up processing specified in the FT request and attempts to execute the resulting command.
For example, if this option is specified as *-lss=␣file1.txt* and the request specifies *lpr* as follow-up processing, FTAC executes *lpr␣file1.txt* as follow-up processing.

Prefix and suffix and follow-up processing command must together not be longer than 1000 bytes (for the representation in UTF-8, see page 163).

Please also bear in mind the information provided on the *-ls* option!

If a suffix was defined with *-lss*, the character set available for specifying follow-up processing in the FT request is restricted to:
– alphanumeric characters (letters and digits)
– the special characters + = / ! ␣ - , @ _ " $ '
– a period (.) between alphanumeric characters

-lss not specified

FTAC adds no suffix to the follow-up processing specified in the request in the event of successful file transfer.

-lf=command4 | **@n**

-lf specifies follow-up processing to be executed under your login name if the file transfer is aborted due to an error. If *-lf* is specified, no failure follow-up processing may be requested in the FT request. Making an *-lf* entry only makes sense if you also make an entry for *-ls* (see above) to preclude the possibility that a successful request can circumvent the *-lf* entry. If you have defined a prefix for the file name with *-fnp* and plan follow-up processing for this file, you must specify the complete file name here.

@n for *command4*

If *-lf=@n* is specified, no failure follow-up processing is then permitted in the event of unsuccessful file transfer.

-lf not specified

does not restrict follow-up processing in the local system in the event of unsuccessful file transfer (Exception see *-lfp* or *-lfs*).

-lfp=command5

-lfp defines a prefix for follow-up processing in the local system in the event of unsuccessful file transfer. FTAC then sets the character string *command5* in front of the follow-up processing specified in the FT request and attempts to execute the resulting command.
For example, if this option is specified as *-lfp='lpr␣"* and the request specifies *file2.txt* as follow-up processing, FTAC executes *lpr␣file2.txt* as follow-up processing.
Prefix and suffix and follow-up processing command must together not be longer than 1000 bytes (for the representation in UTF-8, see page 163).

Please also bear in mind the information provided on the *-lf* option!

If a suffix was defined with *-lfs*, the character set available for specifying follow-up processing in the FT request is restricted to:
– alphanumeric characters (letters and digits)
– the special characters + = / ! ␣ - , @ _ " $ '
– a period (.) between alphanumeric characters

-lfp not specified

FTAC sets no prefix in front of the follow-up processing specified in the request in the event of unsuccessful file transfer.

-lfs=command6

-lfs defines a suffix for follow-up processing in the local system in the event of unsuccessful file transfer. FTAC then sets the character string *command6* after the follow-up processing specified in the FT request and attempts to execute the resulting command.
For example, if this option is specified as *-lfs=␣error.txt* and the request specifies *lpr* as follow-up processing, FTAC executes *lpr␣error.txt* as follow-up processing.

Prefix and suffix and follow-up processing command must together not be longer than 1000 bytes (for the representation in UTF-8, see page 163).

Please also bear in mind the information provided on the *-lf* option!

If a suffix was defined with *-lfs*, the character set available for specifying follow-up processing in the FT request is restricted to:
– alphanumeric characters (letters and digits)
– the special characters + = / ! ␣ - , @ _ " $ '
– a period (.) between alphanumeric characters

-lfs not specified
FTAC sets no suffix after the follow-up processing specified in the request in the event of unsuccessful file transfer.

-wm=o | **-wm=n** | **-wm=e** | **-wm=one**
-wm specifies which write modes may be used in the file transfer request and what they effect.

o (overwrite) In the FT request of openFT or FTAM partners, only *-o* or *-e* may be entered for write mode. The receive file is overwritten if it already exists, and is created if it does not yet exist.

With FTP partners, *-n* may also be entered if the file does not yet exist.

n (no overwrite) In the FT request *-o*, *-n* or *-e* may be entered for write mode. The receive file is created if it does not yet exist. If the receive file already exists, the request is not executed.

e (extend) In the FT request only *-e* may be entered for write mode, i.e. the receive file is extended by appending the transferred file to the end if the receive already exists. The receive file is created if it does not yet exist.

one (default value)
means that the FT profile does not restrict the write mode.

-c=y | **-c=n**
Precondition: openFT-CR must be installed.

Using *-c*, you can determine whether data encryption is required or forbidden. If the setting in the profile does not correspond to the setting in the request, the request is denied. The setting is not valid for file management requests, since there is no encryption for these requests.

y Only requests *with* data encryption may be processed using this profile.

n Only requests *without* data encryption may be processed using this profile.

-c not specified
Data encryption is neither required nor forbidden.

-txt=text
: enables you to store a comment in the FT profile (up to 100 characters).

-txt not specified
: the FT profile is stored without a comment.

CAUTION!

If you use the options *-ff=p*, *-fn*, *-fnp*, *-ls* , *-lsp*, *-lss*, *-lf*, *-lfp* or *-lfs*, you must remember

- that a file-name restriction can be bypassed by renaming the file unless follow-up processing is also restricted;
- that follow-up processing must always be restricted for both successful and unsuccessful file transfer and, if necessary, equivalent restrictions must exist for any permitted preprocessing;
- that prefixes for the file name and follow-up processing must be matched to one another;
- that no symbolic links should occur in the part of your file tree that is referenced by the file name prefix.
- that restrictions applied to preprocessing, postprocessing, or follow-up processing can be circumvented if it is possible to replace this command with, for example, a "Trojan horse".

Examples

1. You wish to create an FT profile for the following purpose:

   The Duck Goldmines are to be able to send their monthly reports from their computer *goldmine* to the president at head office via file transfer. The file *monthlyreport_goldmine01* is to be printed out after transfer. The command required to create such an FT profile at head office is:

   ```
   ftcrep␣goldmrep␣fortheboss␣-d=20171231␣-dir=f\
   ␣-pn=goldmine␣-fn=monthlyreport_goldmine01\
   ␣-ls='lpr␣monthlyreport_goldmine01'␣-lf=@n␣-wm=o
   ```

   The FT profile has the name *goldmrep* and the transfer admission *fortheboss*. It permits only the *monthlyreport_goldmine01* file to be transferred to the bank. Following successful transfer, the file is printed out in the bank. Follow-up processing after unsuccessful file transfer is, however, prohibited. The transfer admission is only valid until December 30, 2017, the FT profile disabled as of 00:00 hours on December 31, 2017.

2. You want to set up the standard admission profile on your user ID in such a way that only the file transfer and file creation functions are possible. This profile can, for instance, be used by FTAM partners that always have to specify the user ID and the password for inbound access.

   The command is as follows:

   ```
   ftcrep␣@s␣@n␣-wm=n␣-ff=t
   ```

3. You want to define an admission profile *monitor1* that only allows monitoring data to be output. Assign *onlyftmonitor* as the transfer admission. The command is as follows:

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

   The purpose of the blank after *FTMONITOR is to automatically separate any options specified during the call from the command. A profile such as this can be used to call the openFT monitor (e.g. using the *ftmonitor* command) and in the *ncopy* command. The admission profile is only valid for communicating via the openFT protocol.

   You will find further details in the section “Monitoring with openFT” on page 74.

6.11 ftdeli - Deactivate an instance

The *ftdeli* command allows you to deactivate an instance. Deactivating an instance removes only the symbolic link in the local */var/openFT* directory. The instance file tree is not changed. The standard instance *std* and the currently set instance can not be deleted.

Format

ftdeli -h |
<instance 1..8>

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

instance
Name of the instance to be deactivated. Using the *ftshwi @a* command displays the names of all instances.

Messages of the ftdeli command

If *ftdeli* could not be executed properly, a self-explaining message is output. The exit code is not equal zero in this case.

Examples

1. The instance *inst1* from the directory */CLUSTER/inst1* is to be deactivated on computer *CLUSTER1*, since it has been switched over to *CLUSTER2.* The directory */CLUSTER/inst1* is retained.

```
ftdeli inst1
```

2. Instance *inst2* with the directory */CLUSTER/inst2* is to be deleted along with the instance file tree.

```
ftdeli inst2
rm -r /CLUSTER/inst2
```

3. Using . *ftseti*, it was changed to instance *inst3*. There, an attempt is being made to deactivate the instance *inst3*.

```
ftdeli inst3
ftdeli: openFT Instance 'inst3' can not be removed.
```

6.12 ftdelk - Delete key pair set

You use this command to delete the key pair sets for a reference. Your system can then no longer be authenticated by partner systems which still use the associated public key. For more information on administering keys, see section “Authentication” on page 77.

A key pair set should always be present in your openFT instance as otherwise all requests are run unencrypted, i.e. neither the request description data nor the file contents are encrypted.

Format

ftdelk [-h] <key reference 1..9999999>

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

key reference
: Used to select the key pair set that is to be deleted. You can find the reference in the name of the public key file, see section “Creating and administering local RSA key pairs” on page 80.

6.13 ftdell - Delete log record or offline log file

With *ftdell*, you can delete log records for all login names if you are FT, FTAC or ADM administrator. This function is not permitted for the ordinary user.

You can also delete offline log files that you no longer need. You can delete up to 1024 log files with each *ftdell* command. If you want to delete more files then you must repeat the command.

This command is not permitted for normal users.

Store the log records by redirecting the output of *ftshwl* to a file or to the printer (see section "ftshwl - Display log records and offline log files" on page 297).

Deleting log records changes the size of the file since the storage space is freed immediately after deletion.

The time by which the log records are to be deleted can be entered either as a fixed time with date and time or as a relative time; for example: all records before 10 days ago.

You can also automate the deletion of log records by using the *ftmodo command* to set the corresponding options (*-ld*, *-lda*, *-ldd*, *-ldt*) in the operating parameters. This is recommended if you only want to retain logging information of up to a given age. However, you should not use this method if you want to maintain a continuous long-term archive of the log records

Format

ftdell -h |
[-rg=[[[yyyy]mm]dd]hhmm | -rg=#1..999999999999 | -rg=0..999] |
[-tlf=yyyymmdd[hh[mm[ss]]]]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-rg=[[[yyyy]mm]dd]hhmm
You use *-rg* to specify the end of a logging interval.
When selecting the time, this is interpreted as follows:
- a 4-digit specification is interpreted as the time expressed in hours and minutes,
- a 6-digit specification as the day (date) and time in hours and minutes,
- an 8-digit specification as the month, day, and time in hours and minutes,
- a 12-digit specification as the year, month, day, and time in hours and minutes.

The largest possible value that can be specified as the date is 20380119 (January 19, 2038).
openFT then deletes all log records which are older that the specified time. The optional data ([...]) is automatically replaced by current values.

-rg=#1..999999999999
Here you use *-rg* to specify the end log ID. It is identified by a leading # character, followed by the 1-12-digit ID.
openFT then deletes all log records which belong to this log ID or which belong to a smaller log ID.

-rg=0..999
Here you use *-rg* to specify a time interval (relative to the current date and time) as a multiple of 24 hours, i.e. number of days.
openFT then deletes all log records which are older than the specified time. This means you are looking back in time. If you specify *-rg=2*, for example, all log records which are older than two days (48 hours) are deleted.

-rg not specified
The range is not a selection criterion, i.e. all log records are to be deleted by 00:00 hours of the current date.

-tlf=yyyymmdd[hh[mm[ss]]]
-tlf deletes all the offline log files that were switched offline on or before the specified time (local time!). This ensures that only log files that are at least as old as the specified time are deleted.

You must always specify the date as an 8-digit value indicating the year month and day. The year must be greater than or equal to 2000. You can specify the time (hhmmss) partially or not at all if you wish. "00" is added to replace any missing specifications.

If you enter the current date or a date in the future then all the existing offline logging records are deleted.

The options *-rg* and *-tlf* must not be specified simultaneously!

Examples

1. As the FT or FTAC administrator, you wish to delete all FT log records written up to 00:00 hours of the current date.

   ```
   ftdell
   ```

2. As the FT or FTAC administrator, you wish to delete all FT log records written up to the current time:

   ```
   ftdell -rg=0
   ```

3. As the FT or FTAC administrator, you wish to delete all log records written before the last 7-day period (7 times 24 hours before the current time:

   ```
   ftdell -rg=7
   ```

4. As the FT or FTAC administrator, you wish to delete all log records from the beginning to the record with the log ID 1450:

   ```
   ftdell -rg=#1450
   ```

5. As the FT or FTAC administrator, you want to delete all the offline log files that were set offline before 01.04.2012:

   ```
   ftdell -tlf=20120331235959
   ```

6.14 ftdelp - Delete FT profiles

ftdelp stands for "delete profile". You should occasionally thin out the set of profiles (with *ftshwp*) to ensure that no out-of-date admission profiles are retained that could potentially threaten the security of your system.

ftdelp allows the FTAC administrator to delete FT profiles belonging to other login names as well.

ftdelp allows the ADM administrator to delete ADM profiles (i.e. FT profiles with the property "access to remote administration server").

Format

```
ftdelp  -h  |
          <profile name 1..8> | @s | @a
        [ -s=[<transfer admission 8..32> | @a | @n]
             [,<user ID 1..32> | @a | @adm] ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

profile name | **@s** | **@a**
: is the name of the FT profile you wish to delete.

  @s for *profile name*
  : Deletes the standard admission profile for the user ID.

  @a for *profile name*
  : profile name is not used as a criterion for selecting the FT profile to be deleted. If you do not identify the profile more closely with *-s* (see below) you will delete all of your FT profiles.

-s=[transfer admission | **@a** | **@n**][**,**user ID | **@a**]
: *-s* is used to specify criteria for selecting the FT profiles to be deleted.

  transfer admission
  : is the transfer admission of the FT profile to be deleted. A binary transfer admission must be specified in the form x\'...\' or X\'...\' .

  @a for *transfer admission*
  : deletes either the FT profile specified by *profile name* (see above) or all of your FT profiles.

    As the FTAC administrator, you must specify *@a* if you want to delete FT profiles belonging to other login names, since you actually should not know the transfer admission.

@n for *transfer admission*
: As the FTAC administrator, you can specify *@n* if you only want to delete FT profiles of other login names, which do not have any defined transfer admissions.

transfer admission not specified
: causes to query the transfer admission on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the transfer admission. To exclude the possibility of typing errors, the program prompts you to enter the transfer admission a second time. If you just press <ENTER>, this has the same effect as specifying *@a*.

,user ID
: As the FTAC administrator, you can specify any login name.

@a for *user ID*
: If you specify *@a* as the FTAC administrator, FT profiles belonging to all login names are deleted.

@adm for *user ID*
: If you specify *@adm* as the FTAC or ADM administrator, ADM profiles are deleted.

user ID not specified
: deletes only profiles belonging to the user's own login name, regardless of who issues the command.

-s not specified
: if *@a* is specified for *profile name*, all the FT profiles belonging to the login name under which the *ftdelp* command is issued are deleted. Otherwise, the FT profile with the specified name is deleted.

6.15 ftexpc - Export the configuration of the remote administration server

ftexpc stands for "export configuration". If you are the administrator of the remote administration server (= ADM administrator), *ftexpc* allows you to export the configuration data of the remote administration server into an XML file. The content of the XML file with the exported configuration is encoded using UTF-8.

You can use *ftexpc* if you wish to change an existing configuration. To do this, export the existing configuration into an XML file with *ftexpc*, adapt the file (see the section “Creating a configuration file using the Configuration Editor” on page 121 and section “Creating a configuration file using a text or XML editor” on page 124) and then import the changed file again with *ftimpc*.

Format

ftexpc -h |
<file name 1..512>

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

file name
specifies the name of the XML file in which the exported configuration data is to be saved.

The file is created by the *ftexpc* command and must not exist beforehand.

Messages of the ftexpc command

If ftexpc could not be executed properly, a self-explaining message is output. The exit code is not equal zero in this case

6.16 ftexpe - Export FT profiles and admission sets

ftexpe stands for "export environment", i.e. exporting the FTAC environment, or exporting FT profiles and admission sets.

Using *ftexpe* the FTAC administrator can write FT profiles and admission sets of any login names to files, thereby saving them.

However, the standard admission set is not saved and the variable values in an admission set (values marked with an asterisk (*)) that refer to the standard admission set, are saved as variables. This means that there is no fixed value for the relevant basic function in the backup. If an admission set is imported, the relevant basic function receives the value of the standard admission set that is currently valid.

FT profiles and admission sets saved in this way can be re-imported using the *ftimpe* command.

The timestamp of an admission profile is not changed on an export or import operation.

Format

```
ftexpe  -h  |
          <file name 1..512>
        [ -u=<user ID 1..32>[,...,<user ID(100) 1..32>] ]
        [ -pr=<profile name 1..8>[,...,<profile name(100) 1..8>] | -pr=@n ]
        [ -as=y | -as=n ]
        [ -adm=y | -adm=n ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

file name
: With *file name*, you specify the name of the file in which the FT profiles and records are to be written. You may access this file only using the *ftimpe* and *ftshwe* commands. *file name* must not be longer than 512 characters, and no file with the same name must exist in the directory.

-u=user ID1[**,**user ID2][**,**user ID3]...
: *-u* specifies the login names whose FT profiles and admission sets are to be saved to a file. Up to 100 login names can be specified simultaneously.

-u not specified
: all FT profiles and admission sets on the system are saved to the specified file.

-pr=profile name1[**,**profile name2][**,**profile name3]... | **@n**
specifies the FT profiles to be saved to the specified file (up to 100).

@n for *profile name*
no FT profiles are saved.

-pr not specified
all FT profiles belonging to the login names specified in the -u parameter, are saved.

-as=y | **-as=n**
specifies whether or not the admission sets should be saved to the specified file. Possible values are:

y (default value)
all admission sets belonging to the login names specified in the *-u* parameter, are saved.

n no admission sets are saved.

-adm=y | **-adm=n**
specifies whether or not the ADM profiles (i.e. FT profiles with the property "access to remote administration server", corresponding to *ftcrep -ff=c)* should be saved to the specified file. Possible values are:

y (default value)
all ADM profiles are saved.

n no ADM profiles are saved.

Example

The admission set and the FT profiles belonging to the login name *donald* are to be saved. *ftacsave* is specified for the backup file.

```
ftexpe␣ftacsave␣-u=donald
```

6.17 fthelp - Display information on the log record reason codes

With *fthelp*, you can have the meanings of the reason codes for the log function displayed on the screen (RC column in *ftshwl* output).

You can also request the output of the message texts associated with the exit codes of certain FT commands.

Format

fthelp -h | <number 1..ffff>

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

number

This is a four-digit reason code from the log function or the exit code of an FT command belonging to a synchronous FT request. The reason code contains encoded information on an FT request accepted by openFT.

The reason codes and their meanings are listed in the section “Reason codes of the logging function” on page 320.

The exit codes (= message numbers) are listed in section “Exit codes and messages for administration commands” on page 437.

Example

You wish to find out the meaning of reason code 3001.

```
fthelp␣3001

3001 Request rejected. Invalid user identification.
```

Thus, reason code 3001 means that the specified login name or transfer admission is invalid.

6.18 ftimpc - Import the configuration of the remote administration server

ftimpc stands for "import configuration". If you are an ADM administrator, *ftimpc* allows you to import an XML file containing configuration data on the remote administration server. The existing configuration is overwritten on import.

The format of the XML file must match the format in the schema defined in *config.xsd*. *config.xsd* is located in the openFT installation directory under the directory *include*. You will find further details on creating a configuration file in the section “Creating a configuration file using a text or XML editor” on page 124 and section “Creating a configuration file using the Configuration Editor” on page 121.

The XML file is checked for correct syntax and semantics by the XML parser and XML schema validator during import. If errors occur, a message is output to *stderr* indicating the element or the row/column in which the error occurred. The messages generated always appear in English.

In some cases, it is possible that you will receive a message during import indicating that the configuration data cannot be imported and that the asynchronous openFT server must be terminated. In this case, stop the asynchronous openFT server (e.g. using the *ftstop* command), call the *ftimpc* command again and then restart the asynchronous openFT server (e.g. using the *ftstart* command).

You can use *ftimpc* if you wish to change an existing configuration. To do this, export the existing configuration into an XML file with *ftexpc*, adapt the file and then import the changed file again with *ftimpc*.

The content of the XML file exported with *ftexpc* is encoded using UTF-8 (see the section “ftexpc - Export the configuration of the remote administration server” on page 209). You should therefore also encode an import file in UTF-8.

Format

ftimpc -h |
<file name 1..512>

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

file name
specifies the name of the XML file to be imported.

Messages of the ftimpc command

If ftimpc could not be executed properly, a self-explaining message is output. The exit code is not equal zero in this case.

6.19 ftimpe - Import profiles and admission sets

ftimpe stands for "import environment", i.e. importing the FTAC environment or importing FT profiles and admission sets. Using *ftimpe*, the FTAC administrator can import the FT profiles and admission sets of any login names from a file that was created using the *ftexpe* command.

Only those FT profiles whose profile names have not been specified for other FT profiles under the specified login name are imported.

If a profile with the same name is already present, the timestamp (LAST-MODIF with *ftshwp -l*) indicates which has the most recent status.

An FT profile whose transfer admission has already been defined for another FT profile in the system will be imported, but has an undefined transfer admission. It must therefore be assigned a new transfer admission using the *ftmodp* command before it is used. If the existing FT profile in the system is designated as private, it is immediately disabled. It must be assigned a new transfer admission using the *ftmodp* command, before it is used.

The imported FT profiles are automatically locked and must be unlocked before use with the command *ftmodp* and the parameter *-v=y* if the FTAC administrator does not have FT administrator privileges. Privileged FT profiles lose their privileged status when imported. The FTAC administrator can control this behavior with the *-sec* option provided that he has FT administrator privileges.

The standard admission set is not saved when it is exported. Therefore, the standard admission set on the computer at the time of importing remains valid. Variable values in the imported admission sets, that refer to the standard admission set (and are therefore marked with an asterisk (*), are assigned the value of the standard admission set that is currently valid.

Format

```
ftimpe -h  |
          <file name 1..512>
        [ -u=<user ID 1..32>[,...,<user ID(100) 1..32>] ]
        [ -pr=<profile name 1..8>[,...,<profile name(100) 1..8>] | -pr=@n ]
        [ -as=y | -as=n ]
        [ -sec=s | -sec=h ]
        [ -adm=y | -adm=n ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

file name
: *file name* specifies the file from which the FT profiles and admission sets are to be imported.

-u=user ID1[**,**user ID2][**,**user ID3]...
: specifies the login names whose FT profiles and admission sets are to be imported. You can specify up to 100 login names simultaneously.

-u not specified
: all FT profiles and admission sets are imported.

-pr=profile name1[**,**profile name2][**,**profile name3]...| **-pr=@n**
: specifies the FT profiles to be imported (up to 100).

  @n for *profile name*
  : no FT profiles are imported.

-pr not specified
: all FT profiles belonging to the login names specified in the *-u* parameter are imported. However, the profile is not imported if another FT profile of the same name already exists under this login name.

as=y | **-as=n**
: specifies whether or not admission sets are to be imported. Possible values are:

  y (default value)
  : all admission sets belonging to the login names specified in the *-u* parameter are imported.

  n no admission sets are imported.

-sec=s | **-sec=h**
: *-sec* specifies the security level when importing FT profiles. It only makes sense to use the *-sec* option if you, the FTAC administrator, have FT administrator privileges.

  s (standard) If you have FT administrator privileges, the attributes of the FT profile are not changed when it is imported.

  If you do not have FT administrator privileges, the effect is the same as *-sec=h*, i.e. the profiles are locked.

  -sec=s is the default value.

  h (high) The FT profiles are locked (LOCKED (by import)) and are assigned the attributes *private* and *not privileged*.

-adm=y | **-adm=n**

specifies whether or not the ADM profiles (i.e. FT profiles with the property "access to remote administration server", corresponding to *ftcrep -ff=c)* are to be imported. Possible values are:

y (default value)
: all ADM profiles are imported. This option is permissible only if an ADM administrator is configured on the target computer.

n
: no ADM profiles are imported.

Example

The admission set and FT profiles of the login name *donald* were saved to the file *ftacsave* with *ftexpe*. They are to be imported to another system under the same login name.

```
ftimpe␣ftacsave␣-u=donald
```

As the FTAC administrator you may receive the following messages, for example:

```
OWNER    NAME
donald   secret1     FT profile already exists.
         secret2
```

These messages indicate that *donald* has already created the FT profiles *secret1* and *secret2* on the new system, and these profiles were therefore not imported.

Note

If, after import, you wish to delete an admission set for a login name that does not exist on your computer, enter the command *ftmoda ␣login-name ␣-ml=s*. This situation can occur when you use *ftexpe* to incorporate into your system a file that has been created on a different host.

6.20 ftimpk - Import RSA key

You can use the command *ftimpk* (import key) as FT administrator to import a partner's public key or an RSA key pair from a file. The file is made available by the party that generated the key/RSA key pair. On import, the partner key or RSA key pair is saved at the "correct" location in the openFT instance directory and can then be used for authentication.

Importing public keys of a partner

If you want to import the public key of a partner then this partner must be entered in the partner list. The key is stored in the *syskey* subdirectory with the partner ID as file name.

Importing RSA key pairs

You can import an RSA key pair consisting of a public and a private key. The key pair can be used like a key generated by openFT for data encryption and authentication.

The key pair can have been generated using an external tool. Keys must have the length 768, 1024 or 2048 bit. The keys may be present in PEM format (native PEM or PKCS#8 format without password phrase or, after v1 / v2, with password phrase) or in PKCS#12 V1.0 format.

If the key pair demands a password phrase (password) then this must be specified during the import.

During import, the same applies as for key pairs generated with *ftcrek*:

- The key pair contains a unique reference number.
- The public key is stored under the name **syspkf.r<key-reference>.l<key-length>** in the *config* directory of the openFT instance's instance file tree.

For details, see section “Creating and administering local RSA key pairs” on page 80.

Format

```
ftimpk -h |
        [ -pr=<file name 1..512> ]
        [ -pu=<file name 1..512>]
        [ -p=<password 1..64> | -p= ]
        [ -p12 ]
```

Description

-h Outputs the command syntax on screen. Any specifications after *-h* are ignored.

-pr=file name
(private) indicates that a private and public key are to be imported. *file name* is the absolute or relative path name of the file containing the two keys.

-pu=file-name
(public) indicates that only a public key is to be imported. *file name* is the absolute or relative path name of the file containing the key.

You must always specify either *-pr* or *-pu*!

-p=password | **-p=**
Specifies the password if the key or keys is (are) password-protected.

No password specified
If you specify *-p=* without a password, the password is queried on screen after the command has been sent. The entry you make is not displayed, in order to prevent unauthorized persons from seeing the password.

-p not specified
The password(s) is/are not password-protected, default value.

-p12 The key file contains a certificate and a private key in accordance with the standard PKCS#12 V1.0. The file is searched for a private key and any non-supported elements (e.g. certificates, CRLs) are ignored during the import. The first private key that is found in the file is imported. Any others are ignored.

If the certificate is protected by a signature or hash then openFT does not perform a validity check. The validity of the file must be verified using other means.

-p12 not specified
The private key is not present in PEM format, default value.

Examples

1. You want to import the public key from the file clientkey1 (without a password).

```
ftimpk -pu=clientkey1
```

2. You want to import an RSA key in PEM format that was generated using a tool from the file rsakeys20120303. The keys are protected by a password which you must enter invisibly (hidden) at the screen.

```
ftimpk -pr=rsakeys20120303 -p=
```

6.21 ftlang - Change default language setting

The default language for openFT is determined by evaluating the LANG environment variable during installation (Linux, Solaris, AIX) or, in HP-UX, is set to English by default..

You can switch languages later on using the shell procedure */opt/openFT/bin/ftbin/ftlang*. For more details see section “Switching the language interface” on page 61.

Format

ftlang [-h |
-i |
de |
en]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-i you can use this switch to query the currently set language variant.

de openFT is switched to German as the default.

en openFT is switched to English as the default.

In both cases, the necessary messages files, the *fthelp* procedure, the man pages (Solaris, AIX and HP-UX) and the openFT Explorer including the help texts for the selected language are activated.

Example

1. Check which language is selected:

```
/opt/openFT/bin/ftbin/ftlang –i
en
```

2. The default language setting is switched from German to English:

```
/opt/openFT/bin/ftbin/ftlang␣en
```

6.22 ftmoda - Modify admission sets

ftmoda stands for "modify admission set".

As the FTAC administrator, you can use this command to define settings for the standard admission set and for any admission set of any user in the system. The settings made by the administrator for other users are the MAX. ADM LEVELS.

You can assign a security level of between 0 and 100 for each basic function. These values have the following meanings:

0 The basic function is locked, i.e. it is not released for any partner system.

1 to **99** The basic function is only released for partner systems with the same or a lower security level. You can use the *ftshwptn* command to display the security level of a partner system.

100 The basic function is available for all partner functions.

For basic functions, consult the table on page 223.

The FTAC or ADM administrator can also use *ftmoda* to transfer the FTAC administrator privileges or the ADM administrator privileges to other user IDs.

Format

```
ftmoda -h |
        [ <user ID 1..32> | @s ]
        [ -priv=y ]
        [ -admpriv=y ]
        [ -ml=s | -ml=0..100 ]
        [ -os=s | -os=0..100 ]
        [ -or=s | -or=0..100 ]
        [ -is=s | -is=0..100 ]
        [ -ir=s | -ir=0..100 ]
        [ -ip=s | -ip=0..100 ]
        [ -if=s | -if=0..100 ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

user ID | **@s**
As the FTAC administrator, you can specify any login name desired.

@s for *user ID*
By entering the value @s, the FTAC administrator can modify the standard admission set.

user ID not specified
modifies the admission set of the login name under which *ftmoda* is entered.

-priv=y
As the FTAC administrator, you can assign administrator privileges to the specified *user ID*.

-priv not specified
does not change the FTAC administrator.

-admpriv=y
If you are an ADM administrator, this specification allows you to pass the administration admission for the remote administration server to the *user ID* specified. In addition, all profiles defined with *-ff=c* are forwarded to the new user ID. If profiles with the same name already exist under the new user ID, the command is rejected. If there does not yet exist an ADM administrator on the remote administration server, the FTAC administrator has to define the ADM administrator **first** using *-admpriv=*. Otherwise the remote administration server cannot be administrated, i.e. the configuration file cannot be imported by means of *ftimpc,* for example.

-admpriv not specified
does not change the ADM administrator.

-ml=s | **-ml=**0..100
sets the same value for all six basic functions.

Possible values are:

s sets each of the basic functions to the value defined in the standard admission set.

0 disables all of the basic functions.

1 to **99**
All basic functions are released only for partner systems whose security level is equal to or lower than the specified value.

100 All basic functions are released for all partner systems. For outbound file management functions, no check is made.

-ml not specified
: leaves the settings in the admission set unchanged if none of the following entries are made.

-os=s | **-os=**0..100
: sets the value for the basic function *outbound send,* see page 224 for possible values. *outbound send* means that requests initiated in your local system send data to a remote system.

-or=s | **-or=**0..100
: sets the value for the basic function *outbound receive,* see page 224 for possible values. *outbound receive* means that requests initiated in your local system fetch data from a remote system.

-is=s | **-is=**0..100
: sets the value for the basic function *inbound send*, see page 224 for possible values. *inbound send* means that a remote partner system fetches data from your local system.

-ir=s | **-ir=**0..100
: sets the value for the basic function *inbound receive*, see page 224 for possible values. *inbound send* means that a remote partner system sends data to your local system.

-ip=s | **-ip=**0..100
: sets the value for the basic function *inbound follow-up processing* + *preprocessing* + *postprocessing*, see page 224 for possible values. This determines whether or not a remote system may request follow-up, pre- or postprocessing on your local system.

-if=s | **-if=0**..100
: sets the value for the basic function *inbound file management*, see page 224 for possible values.

  Please note that subcomponents of *inbound file management* are affected by other settings, see “Dependencies concerning inbound file management” on page 224.

-os, *-or, -is, -ir, -ip* or *-if* not specified
: leaves the setting for the respective basic function unchanged.

Possible values for the basic functions

The following values are possible for the individual basic functions (*-os*, *-or*, *-is*, *-ir*, *-ip* and *-if*):

s The specifications in the default admission record apply to the basic functions.

0 The basic function is locked.
With some basic functions, this can also affect inbound file management components. For details, refer to the table on page 224 .

1 to **99**
The basic function is only released for partner systems on which the security level is less than or equal to the specified value.

100 The basic function is released for all partner systems.

Dependencies concerning inbound file management

The subcomponent *Display file attributes* is controlled by the basic *function inbound send*. In addition, the following dependencies on other on other settings exist for some components:

Inbound file management function	**Values of the admission set or extension in profile**
Display file attributes	Inbound Send (IBS) enabled
Modify file attributes	Inbound Receive(IBR) **and** Inbound File Management(IBF) enabled
Rename files	Inbound Receive(IBR) **and** Inbound File Management(IBF) enabled
Delete files	Inbound Receive(IBR) enabled **and** Write mode = overwrite in profile
Display directories	Inbound File Management(IBF) enabled
Create, rename and delete directories	Inbound File Management(IBF) enabled **and** direction = from partner in profile

6.23 ftmodi - Modify an instance

The *ftmodi* command allows you to assign another Internet host name address to an instance.

Note on using more than one instance

All instances must be explicitly assigned their own host name
(option -addr with ftmodi or ftcrei). This also applies to standard instances.

Format

ftmodi -h | <instance 1..8> [-addr=<host name> | -addr=@n]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

instance
: Name of the instance to be modified. Instance names have a maximum length of 8 characters and must consist of alphanumeric characters. The first character must not be a number.

-addr=host name | **-addr=@n**
: Internet host name whose assigned IP address is used to address the instance externally (destination address) and which is used as the sender address with outgoing connections. Changing *-addr* does not affect the instance's operating parameters *instance ID* and *processor*.

  host name
  : A particular or another Internet host name can be assigned to the instance here.

  @n for *host name*
  : This specification is only permitted for the standard instance *std*.

    The standard instance is not assigned a particular host address anymore, and therefore it signs on for all addresses of the system.

    In this manner you can switch from an operation with several instances to a one instance operation.

Examples

1. The host with the name MAPLE is assigned to the default instance. Local requests to 127.0.0.1 are thus no longer possible.
   The command is as follows:

   ```
   ftmodi std -addr=MAPLE
   ```

2. The default instance is to log in with all IP addresses of a system again and listen to all addresses. The command is as follows:

   ```
   ftmodi std -addr=@n
   ```

Messages of the ftmodi command

If *ftmodi* could not be executed properly, a self-explaining message is output. The exit code is not equal zero in this case.

6.24 ftmodk - Modify RSA key

You can use the *ftmodk* command to modify the expiration date and authentication level of keys that are used for the authentication of partner systems. The changes are stored in the relevant key file.

Once the expiration date of a key has been reached, authentication using this key is rejected. However, you can still modify the expiration date after the key's validity has expired, e.g. in order to temporarily re-enable so that a current key can be transferred securely.

Format

ftmodk –h |
[-id=<identification1..64> | -id=@a] |
[-pn=<partner 1..200> | -pn=@a]
[-al=1 | -al=2]
[-exp=[yyyymmdd]]

Description

-h Outputs the command syntax on screen. Any specifications after *-h* are ignored.

-id=identification | **-id=@a**
identification is the instance identification of the partner whose key is to be modified.
-id must not be specified in combination with *-pn*.

@a The installed keys of all partner systems are modified.

-pn=partner | **-pn=@a**
partner is the name of the partner system in the partner list or the address of the partner system whose key is to be modified.
-pn must not be specified in combination with *-id*.

You will find detailed information on address specifications in the section “Specifying partner addresses” on page 68.

@a The installed keys of all partner systems are modified.

Neither *-id* nor *-pn* specified
The installed keys of all partner systems are modified.

-al=1 | **-al=2**

(authentication level) Specifies the authentication level for the key or keys.

1 The authentication level for the partner or partners is set to 1. This corresponds to the possibilities available up to openFT V11.0A.

If the partner system is subsequently authenticated at level 2 then the entry AUTHENTICATION-LEVEL=2 is automatically recorded in its key file.

2 The partner system supports the level 2 authentication procedure introduced in openFT V11.0B. Level 1 authentication attempts are rejected.

-al not specified

The authentication level is unchanged.

-exp=[yyyymmdd]

Specifies the expiration date of the key or keys.

yyyymmdd

Expiration date in the format yyyymmdd, e.g. 20121231 for 31.12.2012. The key or keys can be used for authentication at the latest up until 00:00 on the specified date.

No date specified

exp= without a date specification means that there is no expiration date for the key or keys.

-exp not specified

The expiration date of the key or keys is unchanged.

6.25 ftmodo - Modify operating parameters

You can use *ftmodo* to define and modify the following parameters for openFT operation:

- the key length of the RSA key
- the maximum values for file transfer
- the identification and the name of the local system
- the default value for the security level
- the mode for sender verification
- the logging scope (traces, logging, console traps and ADM traps)
- the automatic deletion of log records
- the switch-over of the log file and trace file
- the scope of measurement data recording
- the variant of the used code table
- the addresses for the individual protocols
- the settings for the remote administration server
- the use of TNS and CMX
- the settings used for user data encryption

For FTAM operation, you can also activate or deactivate the Application Entity Title (AET).

You can also use the openFT Explorer to modify the operating parameters (exception: deactivation of the application entity title).

Format

```
ftmodo -h |
        [ -kl=0 | -kl=768 | -kl=1024 | -kl=2048 ]
        [ -tu=<transport unit size 512..65535> ]
        [ -pl=1 | -pl= ]
        [ -pl=<process limit 1..32> | -pl= ]
        [ -cl=<connection limit 1..255> ]
        [ -admcl=<connection limit 1..255> ]
        [ -admcs=n | -admcs=y ]
        [ -rql=<maximum number of requests 2..32000> ]
        [ -rqt=<request lifetime 1..400> | -rqt= ]
        [ -id=<identification 1..64> ]
        [ -p=<processor name 1..8> ][ -l=<station name 1..8> ]
        [ -sl=<security level 1..100> | -sl=p ][ -ptc=i | -ptc=a ]
        [ -lf=c ][ -lt=a | lt=f | lt=n] [ -lc=a | -lc=m | -lc=r ]
        [ -la=a | -la=f | -la=m | -la=n ]
        [ -ld=n | -ld=f ][ -lda=<0..999> ][ -ldt=hhmm ]
        [ -ldd=@d | Mo | Tu | We | Th | Fr | Sa | Su | <1..31> ]
        [ -mon=n | -mon=f ][ -monr=[l|r][a|s] ]
        [ -monp=a | -monp=[openft][,][ftam][,][ftp] ]
        [ -tr=n | -tr=f | -tr=c ]
        [ -trp=a | -trp=[openft][,][ftam][,][ftp][,][adm] ]
        [ -trr=[ | |r][a | s] ][ -tro=[b] ][ -troll=[s | d]
        [ -atpsv=[<partner 1..200>][,][<transfer admission 8..67> | @d ]
        [ -atp=a | -atp=n | -atp=[[-]fts],[[-]rqs],[[-]rqc],
              [[-]rqf],[[-]pts],[[-]ptu] ]
        [ -tpc=a | -tpc=n | -tpc=[[-]sss],[[-]fts],
              [[-]rqs],[[-]rqc],[[-]rqf],[[-]pts],[[-]ptu] ]
        [ -ccs=<CCS name 1..8> ]
        [ -acta=a | -acta=[openft][,][ftam][,][ftp][,][adm] ]
        [ -ftp=<port number 1..65535> | -ftp=@s]
        [ -openft=[<port number 1..65535>][.<T-Sel 1..8>] |
          -openft=@s ]
        [ -ftam=[<port number 1..65535>][.<T-Sel>[.<S-Sel>[.<P-Sel>]]] |
          -ftam=@s ]
        [ -adm=<port number 1..65535> | -adm=@s ]
        [ -ftstd=<port number 1..65535> | -ftstd=@s ]
        [ -tns=y | -tns=n ]
        [ -cmx=y | -cmx=n ]
        [ -ae=y | -ae=n ]
        [ -dp=n | -dp=f ]
        [ -c= | -c=i | -c=o | -c=io | -c=oi ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-kl=0 | **-kl=768** | **-kl=1024** | **-kl=2048**
The *-kl* parameter can be used to change the length of the RSA key used in encryption. The value of the *kl* parameter specifies the new RSA key length in bits. The RSA key is only used for the encryption of the AES key agreed between the partners (or for encrypting the DES key in versions up to openFT V7.0).
openFT uses the AES key for encrypting request description data and any file content present.
The *ftmodo -kl=*... command can be specified in current openFT operation.

-kl=0 explicitly deactivates encryption. If this is set during operation then any requests with encryption (prior to *ftmodo -kl=0*) that have been submitted but not yet started are aborted with errors. Any running requests are processed and their encryption is retained. New requests using encryption are rejected.

After reinstallation, the default value *-kl=768* is used.

Default setting following initial installation: *-kl=2048*.

-tu=transport unit size
You use the parameter *-tu* to define the upper limit for message length at transport level (block length). You can choose a value between 512 and 65535.

The block length only applies to requests to openFT partners.

Default setting following initial installation: *-tu=65535*.

-pl=1 | **-pl=**
Maximum number of processes used for the processing of asynchronous requests.

1 All asynchronous requests are processed by the same process.

No value specified
If you specify *-pl=* without parameters then the number of processes is equal to the number of connections, i.e. each connection is handled by a separate process.

Default setting following initial installation: *-pl=* (i.e. no number specified).

pl=process limit | **-pl=**
process limit is the maximum number of openFT servers used for the processing of asynchronous requests.

process limit not specified
If you specify *-pl=* without parameters then the number of openFT servers is equal to the number of connections, i.e. each connection is handled by a separate openFT server.

-cl=connection limit

Maximum number of asynchronous requests that are processed simultaneously. Possible values: 1 to 255.

The default value is 16.

Default setting following initial installation: *-cl=16*.

-pl=2 means that a maximum of two openFT servers are used to process asynchronous requests. *-cl=16* means that a maximum of 16 requests can be processed simultaneously. However, this means that the second openFT server is not started until the first openFT server has reached its assigned limit of 8 connections! This value is calculated by dividing the value of *-cl* by the value of *-pl*.

-admcl=connection limit

Maximum number of connections provided for remote administration requests. Possible values: 1 through 255.

Read the note under -*admcs*.

Default setting following initial installation: *-admcl=8*.

-admcs=n | **-admcs=y**

Specifies whether the local openFT instance is flagged as a remote administration server.

y Flags the local openFT instance as a remote administration server. This means that this instance can also be an ADM trap server.

n The local openFT instance is not (no longer) flagged as a remote administration server. This means that it is not (no longer) possible to receive ADM traps. This is the default after a new installation.

If you specify *-admcs*, but do not specify *-admcl*, then openFT sets the connection limit (*-admcl)* to the following value:

64 if *-admcs=y*.

8 if *-admcs=n*.

Default setting following initial installation: *-admcs=n*.

rql=maximum number of requests

You use *-rql* to specify the maximum number of entries in the request queue. You can choose a value between 2 and 32000.

Default setting following initial installation: *-rql=2000*.

-rqt=request lifetime | **-rqt=**

You use *-rqt* to specify the maximum lifetime of requests in the request queue. The value applies to both inbound and outbound requests and is specified in days. Values between 1 and 400 are permitted. Once the specified period has expired, requests are deleted from the request queue.

request lifetime not specified:

If you specify *-rqt=* without parameters then the maximum lifetime is unlimited.

Default setting following initial installation: *-rqt=30*.

-id=identification

Specifying the instance identification of your instance. Partner systems using openFT Version 8.1 and later, address your system via this string. In return, openFT uses the instance ID as the sender address when addressing the partners. The instance ID must be unique and not case-sensitive (see also section “Instance Identifications” on page 79). If you modify the instance ID, the relevant public key files will be automatically updated.

Default setting following initial installation: *-id= local DNS name or host name*.

-p=processor name

You specify the processor name assigned to your system here.

No processor name is specified after initial installation.

-l=station name

The station name of the openFT application. The default value is $FJAM.

Default setting following initial installation: *-l=$FJAM*.

The specifications for *processor name* and *station name* depend on how your system is connected to the network. Further details can be found in the chapter “Installation” on page 25openFT System Administrator Guide.

-sl=security level | **-sl=p**

You use this option to define the default security level. This level applies to partners in the partner list to which no explicit security level value was assigned when they were entered with *ftaddptn*. The effect also depends on the settings for the admission set, see the *ftmoda* command on page 221.

security level

Specifies a fixed default security level. Values between 1 and 100 are permitted. 1 indicates a very low and 100 a very high requirement for protection with regard to the partners.

p The default security level depends on the partner's attributes:

- Security level 10 if the partner has been authenticated.
- Security level 90 if the partner is known in the transport system and is identified by the name it is known by in the transport system.
- Security level 100 otherwise, i.e. if the partner has only been identified by its address.

Default setting following initial installation: *-sl=p*.

-ptc=i | **-ptc=a**

This allows you to modify the global settings for sender verification. This setting only applies to named partners which are connected via the openFT protocol and do not use authentication. In the case of dynamic partners, as well as for FTAM and FTP partners, this setting has no effect.

i (identification)
Deactivates verification of the transport address. Only the identification of the partner is checked.

a (address)
Activates verification of the transport address
If the transport address under which the partner logs in does not correspond to the entry in the partner list then the request is rejected.

Default setting following initial installation: *-ptc=i*.

-lf=c The log file is changed.

The new log file is created under the name *syslog.Lyymmdd.Lhhmmss*:
– *yymmdd* is the date (year, month, day) on which the file was created,
– *hhmmss* is the time (hour, minute, second for GMT) on which the file was created.

-lt=a | **-lt=f** | **-lt=n**

This option is used to selectively deactivate FT log records.
Possible values:

a (all)
Log records are written for all FT requests.

f (failure case)
Log records are written for failed FT requests only.

n (none)
No log records are written.

Default setting following initial installation: *-lt=a*.

-lc=a | **-lc=m** | **-lc=r**

This option is used to selectively activate/deactivate FTAC log records.
Possible values:

a (all)
Log records are written for all FTAC access checks.

m (modifying FM calls)
Log records are written for all modifying file management requests leaving the remote system as well as for all rejected FTAC access checks.

r (reject case)
Log records are written for rejected FTAC access checks only.

Default setting following initial installation: *-lc=a*.

-la=a | **-la=f** | **-la=m** | **-la=n**

This option allows you to selectively activate the logging of administrative requests. The following parameters are available:

a (all)
Log records are written for all administration requests.

f (failure)
Log records are only written for failed administration requests.

m (modifying)
Log records are written for all administration requests that make modifications.

n (none)
No log records are written for administration requests.

Default setting following initial installation: *-la=a*.

-ld=n | **-ld=f**

This option allows you to control whether log records are deleted automatically.

n (on) Activates the automatic deletion of log records. This activates the criteria specified in *-lda*, *-ldt* and *-ldd* (minimum age and deletion interval).

f (off) Deactivates the automatic deletion of log records. When this option is set, the settings made for *-lda*, *-ldt* and *-ldd* have no effect.

Default setting following initial installation: *-ld=f*.

-lda=0..999

Minimum age of log records for deletion in days. The days are counted back from the deletion time specified in *-ldt*. The value 0 deletes all the log records that were written before or on the time of the current day specified in *-ldt*.

Default setting following initial installation: *-lda=14*.

-ldt=hhmm

Specifies the (local) time at which the log records are to be deleted. Depending on the system, the delete function may be executed up to 5 minutes before the time specified here.

Default setting following initial installation: *-ldt=0000* (i.e. time = 00:00).

-ldd=@d | **Mo** | **Tu** | **We** | **Th** | **Fr** | **Sa** | **Su** | 1..31

Specifies the day on which the log records are to be deleted.

Mo | **Tu** | **We** | **Th** | **Fr** | **Sa** | **Su**
: Delete every week on the selected weekday (Mo=Monday, .. Su=Sunday).

1..31
: Delete every month on a specific day of the month (1-31). If the value 29, 30 or 31 is specified for a month that has fewer days than this then deletion is performed on the last day of the month.

@d
: The log records are deleted every day.

Default setting following initial installation: *-ldd=@d* (i.e. delete every day).

-mon=n | **-mon=f**

This allows you to activate and deactivate openFT monitoring.

n (on)
: openFT monitoring is activated.

f (off)
: openFT monitoring is deactivated.

Default setting following initial installation: *-mon=f*.

-monr= | **-monr=**[**l**|**r**][**a**|**s**]

This allows you to select openFT monitoring depending on the request type. The value *l* or *r* can be combined with *a* or *s* (Boolean AND, e.g. *la*, *al*, *ls*, *rs*, ...).

l (local)
: Monitoring data is collected for requests issued locally.

r (remote)
: Monitoring data is collected for requests issued remotely.

a (asynchronous)
: Monitoring data is collected for asynchronous requests. Requests issued remotely are always regarded as asynchronous.

s (synchronous)
: Monitoring data is collected for synchronous requests. Synchronous requests are always issued locally.

No request type specified
: If you specify *-monr=*, monitoring data is collected for all requests.

Note that *-monr=rs* does not completely deactivate monitoring. *-monr=rs* has the same effect as *-monp=*. See the section “Description of the monitoring values” on page 324.

Default setting following initial installation: *-monr=*.

-monp= | **-monp=a** | **-monp=**[**openft**][**,**][**ftam**][**,**][**ftp**]
This allows you to select openFT monitoring depending on the protocol type used for the partners. Combinations are also permitted if you specify the protocols individually (separated by commas).

a Monitoring data is collected for all partners.

openft
Monitoring data is collected for openFT partners.

ftam Monitoring data is collected for FTAM partners.

ftp Monitoring data is collected for FTP partners.

No protocol type specified
If you specify *-monp=* with no parameters, the monitoring is deactivated for partners. In this event, only certain monitoring data values are populated. See the section “Description of the monitoring values” on page 324.

Default setting following initial installation: *-monp=a*

-tr=n | **-tr=f** | **-tr=c**
This allows you to activate and deactivate the openFT trace function.

n (on)
The openFT trace function is activated.

f (off)
The openFT trace function is deactivated.

c (change)
The current trace file is closed and a new one is opened.

Default setting following initial installation: *-tr=f*.

-trp=a | **-trp=**[**openft**][**,**][**ftam**][**,**][**ftp**][**,**][**adm**]]
This allows you to select the openFT trace function depending on the type of protocol used for the partners by specifying a comma-separated list of one or more protocol types. All the partners that are addressed via this or these protocol type(s) are then traced.

You can modify the selection made here on a partner-specific basis, see the *-tr* option in the *ftmodptn* command on page 271.

a (all)
All protocol types, and consequently all partners, are selected for tracing.

openft
All partners addressed via the openFT protocol are selected for tracing.

ftam All partners addressed via the FTAM protocol are selected for tracing.

ftp All partners addressed via the FTP protocol are selected for tracing.

adm All partners addressed via the FTADM protocol are selected for tracing.

No protocol type selected
If you specify *-trp=* without parameters then no partner is selected for tracing. In this case, only those partners for which tracing has been activated on a partner-specific basis using *ftmodptn ... tr=n* are traced, see page 271.

Default setting following initial installation: *-trp=a*.

-trr=[l | **r][a** | **s]**
This option allows you to select the request types that are to be traced. The value *l* or *r* can be combined with *a* or *s* (Boolean AND, e.g. *la*, *al*, *ls*, *rs*, ...).

l (local)
All locally submitted requests are selected for tracing.

r (remote)
All remotely submitted requests are selected for tracing.

a (asynchronous)
All asynchronous requests are selected for tracing. Requests issued remotely are always regarded as asynchronous.

s (synchronous)
All synchronous requests are selected for tracing. Synchronous requests are always issued locally.

No request type specified
If you specify *-trr=* without parameters then all requests are selected for tracing.

Note that *-trr=rs* does not completely deactivate tracing. Interface trace files, for instance, continue to be created (if activated).

Default setting following initial installation: *-trr=*.

-tro=[**b**]

You can use *-tro* to select options for the trace function. These options are only effective if the trace function is active.

b (no bulk data)

Minimum trace. Only protocol elements with no file contents (bulk data) are written to the trace file. In the case of protocol elements with file contents, the trace file simply notes that records have been suppressed at this point. This note is entered only once for a sequence of similar records.

No option specified

If you specify *-tro=* without parameters then the trace is written normally.

Default setting following initial installation: *-tro=*.

-troll=[**s** | **d**]

You use *-troll* to define the scope of the trace for the lower protocol layers. This option is effective only if the trace function is activated.

s (standard) Additional entries are written in the standard scope for the lower protocol layers. The standard scope comprises comprehensive logging of the calls, their arguments, the content of any options and the user data.

d (detail) In addition to the standard scope, internal events and transport system information (e.g. system calls) are written for the lower layers.

No option specified

If you specify *-troll=* with no parameters, no trace is performed for the lower protocol layers.

Note on operation with and without CMX:

- In the case of operation without CMX, the trace entries for the lower protocol layers are written to the openFT trace
- In the case of operation with CMX, CMX traces are generated and stored in the *traces* directory of the associated openFT instance. These can then, for example, also be selected and displayed in the openFT Explorer (*Administration* menu, *Open Trace File* command).
  Using this option, it is therefore possible to activate and deactivate CMX traces during active CMX operation.

Default setting following initial installation: *-troll=*.

-atpsv=[partner][**,**][transfer admission | **@d**]

-atpsv= allows you to specify the settings for the ADM trap server. When you enter the ADM trap server for the first time, you must specify both the partner and the transfer admission. You can subsequently change each of the two parameters individually.

partner
: Name or address of the partner to which the ADM traps are sent. This must either be a name from the partner list or the address must be specified in the form *ftadm://host...*. See the section “Notational conventions” on page 162.

transfer admission
: FTAC transfer admission for accessing the ADM trap server.

@d for *transfer admission*
: If you specify *@d* (blanked), the transfer admission is queried on screen after the command has been sent. Your input is blanked.

neither *partner* nor *transfer admission* specified
: If you specify *-atpsv=* without parameters, you remove the ADM trap server. This means that ADM traps are no longer sent.

Default setting following initial installation: *-atpsv=*.

-atp=a | **-atp=n** | **-atp=**ADM trap list (comma-separated)

-atp allows you to activate and deactivate ADM traps. The ADM trap server to which the ADM traps are to be sent is specified with *-atpsv*.

The following specifications are possible with the *-atp* option:

a (all)
: All ADM traps are written.

n (none)
: No ADM traps are written.

fts Activates the ADM traps on the status of the asynchronous server.

-fts Deactivates the ADM traps on the status of the asynchronous server.

rqs Activates the ADM traps on the status of the request queue.

-rqs Deactivates the ADM traps on the status of the request queue.

rqc Activates the ADM traps when a request has been terminated successfully.

-rqc Deactivates the ADM traps when a request has been terminated successfully.

rqf Activates the ADM traps when a request has failed.

-rqf Deactivates the ADM traps when a request has failed.

pts Activates the ADM traps on the status of the partner system.

-pts Deactivates the ADM traps on the status of the partner system.

ptu Activates the ADM traps if a partner system is not available.

-ptu Deactivates the ADM traps if a partner system is not available.

Default setting following initial installation: *-atp=n*.

-tpc=a | **-tpc=n** | **-tpc=**Console trap list (comma-separated)
You use *-tpc* to activate and deactivate console traps.

In Unix and Windows systems, console traps are written to the openFT file *conslog*. In Unix, BS2000 and z/OS systems they are also output at the console and in Windows systems they are also written to the event log.

For *-tpc* you can enter the following values:

a (all)
All traps are written.

n (none)
No traps are written.

sss Activates traps relating to the status of the openFT subsystem.

-sss Deactivates traps relating to the status of the openFT subsystem.

fts Activates traps relating to the status of the asynchronous server.

-fts Deactivates traps relating to the status of the asynchronous server.

rqs Activates traps relating to the status of the request queue.

-rqs Deactivates traps relating to the status of the request queue.

rqc Activates traps on the successful termination of a request.

-rqc Deactivates traps on the successful termination of a request.

rqf Activates traps on the unsuccessful termination of a request.

-rqf Deactivates traps on the unsuccessful termination of a request.

pts Activates traps relating to the status of partner systems.

-pts Deactivates traps relating to the status of partner systems.

ptu Activates traps when a partner system is inaccessible.

-ptu Deactivates traps when a partner system is inaccessible.

Default setting following initial installation: *-tpc=n*.

-ccs=CCS name

You use *CCS name* to define a new character set which is represented by a code table. This character set is then used as the new default value for transfer requests (*ft, ncopy*). The code table specification is only relevant for requests to openFT partners.

Another character set can be explicitly assigned for *ft* and *ncopy* (options *-lc* and *-rc*).

You can also define your own character set. For details concerning CCS names and the associated code tables, see section “Administering code tables” on page 54the openFT System Administrator Guide.

Default value after new installation: *-ccs=iso88591* (corresponds to ISO8859-1)CP1252

-acta=a | **-acta=**[**openft**][**,**][**ftam**][**,**][**ftp**][**,**][**adm**]

This option allows you to activate or deactivate the asynchronous inbound server. You can activate the asynchronous inbound server for specific protocols (openFT, FTP, FTAM, ADM), by specifying a comma-delimited list of one or more protocol types.

a The asynchronous inbound servers are activated for all installed protocol types.

openft
Activates the asynchronous inbound server for requests via the openFT protocol.

ftam Activates the asynchronous inbound server for requests via the FTAM protocol. A warning is issued if the FTAM protocol is not installed.

ftp Activates the asynchronous inbound server for requests via the FTP protocol. A warning is issued if the FTP protocol is not installed.

adm Activates the asynchronous inbound server for administration requests.

No protocol type specified
Specifying *-acta=* without parameters deactivates all asynchronous inbound servers.

If you specify a list of protocol types then the asynchronous inbound servers of the non-specified protocol types are deactivated!

Default setting following initial installation: *-acta=openft,ftam,adm*.

-ftp=port number | **-ftp=@s**

You use *port number* to specify the port number used by FTP.
Possible values: 1 to 65535.

@s Sets the port number for FTP server to the default value of 21.

Default setting following initial installation: *-ftp=@s*.

-openft=[port number][**.**T-selector] | **-openft=@s**

port number

You can use *port number* to specify a port number other than the default for the local openFT server.

Possible values for the *port number*: 1 to 65535

T-selector

You can also specify a T-selector of between 1 and 8 characters in length. You can specify the selector in printable or hexadecimal format (0xnnnn...). Alphanumeric characters and the special characters # @ $ are permitted for printable selectors. A printable selector will be converted to uppercase, coded in EBCDIC in the protocol and may be padded with spaces internally to the length of eight characters. In this case, the port number and T-selector must be separated by a period.

@s *-openft=@s* sets the port number and the T-selector for the openFT server to the default value, i.e. 1100 and $FJAM.

Please use this function carefully because setting a port number or TNS name other than the default makes it difficult for openFT partners to address the local system!

Default setting following initial installation: *-openft=@s* (d.h. *1100* und *$FJAM*).

Notes on operation with TNS

If you are switching from operation without TNS to operation with TNS (*-tns=y*) and if only the T selector with no port number had previously been set under *-openft*, you must specify the port number explicitly, even if it matches the default value. This is necessary to ensure that the T selector cannot be confused with the global name in the TNS.

For operating with TNS, you can specify a TNS name other than the default for the local openFT server. A period must be placed before the TNS name, e.g. *-openft=.OPNFTSRV.* The TNS name must not contain any period.
The default value for the TNS name is $FJAM.

In the case of operation with TNS, the default value for the TNS name is $FJAM.

-ftam=[port number][**.**T-selector[**.**S-selector[**.**P-selector]]] | **-ftam=@s**

port number

You can use *port number* to specify a port number other than the default for the local FTAM server.

Possible values for the port number: 1 to 65535
The default value for the port number is 4800.

T-selector**.**S-selector**.**P-selector

You can also specify a T-selector, a session selector and a presentation selector, each of which may have a length of 1 to 16 characters. In this case, the port number, T-selector, S-selector and P-selector must be separated by a period. You can specify the selectors in printable or hexadecimal format (0xnnnn...)

T-selectors that start with $FTAM (default value) are coded in EBCDIC and padded with spaces to the length of 8 characters. In the protocol, all other printable T selectors as well as all printable session and presentation selectors are converted to uppercase and coded with variable length in ASCII.

The default value for the TNS name is $FTAM.

S-selectors and *P-selectors* do not have default values because, by default, openFT-FTAM does not use these selectors.

Make sure that you carefully harmonize the specifications for the port number, the transport selector, the session selector and the presentation selector (in this option or in the relevant TNS entry) with your FTAM partners.

@s *-ftam=@s* sets the port number and the TNS name for the FTAM server to the default value, i.e. 4800 and $FTAM.

Default setting following initial installation: *-ftam=@s*.

Notes on operation with TNS

If you switch to operation with TNS again (*-tns=y*) and if only the T selector with no port number had previously been set under *-ftam*, you must specify the port number explicitly, even if it matches the default value. This is necessary to ensure that the T selector cannot be confused with the global name in the TNS.
The default value for the T-selector is $FTAM.

For operating with TNS, you can specify a TNS name other than the default for the local FTAM server. A period must be placed before the TNS name, e.g. *-ftam=.FTAMSERV.* The TNS name must not contain any period.

In the case of operation with TNS, the default value for the TNS name is $FTAM.

-adm=port number | **-adm=@s**

port number allows you to specify the port number used for remote administration.

Possible values: 1 to 65535.

@s *-adm=@s* resets the remote administration port number to the default value of 11000.

Default setting following initial installation: *-adm=@s*.

-ftstd=port number | **-ftstd=@s**

You use *port number* to define the default port number for the addressing of openFT partners via partner addresses.

Possible values: 1 to 65535

Take care when using this option, because when you change the value of the option, openFT partners that use the default openFT port number 1100 can only be accessed if the port number is specified explicitly.

@s *-ftstd=@s* resets the default port number for the addressing of openFT partners via partner addresses. The default port number of 1100 then applies again.

Default setting following initial installation: *-ftstd=@s*.

-tns=y | **-tns=n**

This option allows you to activate or deactivate the use of TNS names. This does not affect the use of TCP/IP host names, IP addresses or partner management, or the explicit specification of the port number and selectors with the *-openft=* and *-ftam=* options.
For operation with TNS to be possible, operation with CMX must be activated (*ftmodo -cmx=y*).

y This activates the use of TNS names for openFT and FTAM transfer.

This is necessary, for example, if other transport protocols are to be used alongside TCP/IP.

n This deactivates the use of TNS names. In this case, it is only possible to use the TCP/IP transport protocol. By default, the port numbers set in the operating parameters are used for communications (options *-openft*, *-ftam* and *-ftstd*).

Caution!
This option should not be changed as long as requests are stored or active. Activation and deactivation of the TNS database can cause the conversion of a partner name to a partner address to change, which could in turn lead to requests failing (above all with restart requests) or to unwanted delivery

of files. After switchover, temporary partner entries can also appear twice in the partner list for a while (see *ftshwptn*), even if the partner name is converted to the same address in both cases.

Default setting following initial installation: *-tns=n*.

-cmx=y | **-cmx=n**

This option allows you to switch between operation with CMX and operation without CMX. You can only perform this switchover if the asynchronous openFT server has not been started You may therefore first have to shut down the asynchronous openFT server, e.g. with *ftstop*.

If you want to work with TNS then operation with CMX must be activated.

y Switches to operation with CMX. This is only possible if CMX is installed in the minimum version required for operation with this openFT version. If CMX is not installed or is not installed in the correct version then the *ftmodo* command is rejected with an error message.

n Switches to operation without CMX.

Default setting following initial installation: *-cmx=n*.

-ae=y | **-ae=n**

This option activates/deactivates the AET (Application Entity Title).

y A "nil Application Entity Title" is included as the calling or called Application Entity Title (AET) for transfer using the FTAM protocol.

n The AET is deactivated. The option only has to be reset to *-ae=n* if FTAM link partners, as responders, do not expect to receive an AET.

Default setting following initial installation: *-ae=y*.

-dp=n | **-dp=f**

You use this option to specify whether or not dynamic partners are permitted.

n (on) Dynamic partners are permitted. Partners can then be accessed via their address irrespective of whether they are entered in the partner list or not.

f (off) Dynamic partners are not permitted, i.e. partners cannot be accessed via their address. As a result, it is only possible to use partners that are entered by name in the partner list and are addressed via the partner name.

Default setting following initial installation: *-dp=n*.

-c= | -c=i | -c=o | -c=io | -c=oi

You use this data to control system-wide user data encryption. This setting applies to transfer requests and administration requests.

i Activates inbound encryption:
Inbound requests must transfer the user data in encrypted form as otherwise they are rejected.

o Activates outbound encryption:
Outbound requests transfer the user data in encrypted form even if no encryption has been specified in the request (e.g. *ft*, *ncopy*, program interface, openFT Explorer).

io, **oi** Activates inbound and outbound encryption:
Inbound requests must transfer the user data in encrypted form as otherwise they are rejected. Outbound requests transfer the user data in encrypted form even if no encryption has been specified in the request.

No encryption option specified
Specify *-c=* to deactivate system-wide user data encryption. If encryption is required then this must be explicitly specified in the request.

- System-wide encryption may only be activated if openFT-CR is installed.
- If inbound encryption is activated then inbound FTAM requests and inbound FTP requests are rejected.
- If outbound encryption is activated then outbound FTAM requests are rejected while outbound FTP requests are permitted.
- File management requests are executed unencrypted irrespective of the setting entered for *-c*.

Default setting following initial installation: *-c=*.

Examples

1. The identification of your own instance is to be set to host.hugo.net:

   ```
   ftmodo -id=host.hugo.net
   ```

2. Only partners from the partner list are to be permitted:

   ```
   ftmodo -dp=f
   ```

3. Flags the local openFT instance as a remote administration server:

   ```
   ftmodo -admcs=y
   ```

4. Only the asynchronous inbound servers for the openFT and FTAM protocols are to be activated.

   ```
   ftmodo -acta=openft,ftam
   ```

6.26 ftmodp - Modify FT profiles

ftmodp stands for "modify profile".

The FTAC administrator can use this command to change or to privilege FT profiles of other users.

The ADM administrator can use this command to change ADM profiles (i.e. FT profiles which have the property "access to remote administration server", corresponding to ***-ff=c***).

The timestamp is updated when a profile is modified.

In the event that the FTAC administrator does not have FT administrator privileges the same time, then admission profiles of other users are blocked after a modification (except after *-priv=y*). This can be by-passed by entering *-ua=user ID,password*. If the user later changes his/her password, the profile will no longer be usable without further modification.

Format

```
ftmodp -h |
          <profile name 1..8> | @s | @a
        [ -s=[<transfer admission 8..32> | @a | @n ]
             [,<user ID 1..32> | @a | @adm] ]
        [ -ua=[ <user ID 1..32> ],[<password 1..20> | @n ] ]
        [ -nn=<profile name 1..8> ]
        [ -tad= | -tad=<transfer admission 8..32> | -tad=@n ]
        [ -v=y | -v=n ]     [ -d=yyyymmdd | -d= ]
        [ -u=pr | -u=pu ]   [ -priv=y | -priv=n ]
        [ -iml=y | -iml=n ]
        [ -iis=y | -iis=n ]   [ -iir=y | -iir=n ]
        [ -iip=y | -iip=n ]   [ -iif=y | -iif=n ]
        [ -ff= | -ff=[t][m][p][r][a][l] | -ff=c ]
        [ -dir=f | -dir=t | -dir=ft ]
        [ -pn=<partner 1..200>,...,<partner(50) 1..200> | -pn= ]
        [ -pna=<partner 1..200>,...,<partner(50) 1..200> ]
        [ -pnr=<partner 1..200>,...,<partner(50) 1..200> ]
        [ -fn=<file name 1..512> | -fn=] [ -fnp=<file name prefix 1..511> ]
        [ -ls= | -ls=@n | -ls=<command1 1..1000> ]
        [ -lsp= | -lsp=[<command2 1..999> ][ -lss= | -lss=command3 1..999> ]
        [ -lf= | -lf=@n | -lf=<command4 1..1000> ]
        [ -lfp= | -lfp=<command5 1..999>][-lfs= | -lfs=<command6 1..999> ]
        [ -wm=o | -wm=n | -wm=e | -wm=one ]
        [ -c= | -c=y | -c=n ]
        [ -txt=<text 1..100> | -txt= ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

profile name
: specifies the name of the FT profile you wish to modify. To see the profile names you have already assigned, you can issue the *ftshwp* command (without options).

@s for *profile name*
: *@s* allows you to change the properties of the standard admission profile of the user ID.

The options *-v*, *-d* and *-u* are ignored with a standard admission profile.

@a for *profile name*
: modifies all FT profiles that come into question at once, unless you select a specific profile with the option *-s*.

If you specify *ftmodp profile name* without any other parameters, you force the timestamp of the profile to be updated.

-s=[transfer admission | **@n** | **@a**][**,**user ID | **@a** | **@adm**]
: is used to specify selection criteria for the FT profile to be modified.

transfer admission
: specifies the transfer admission of the FT profile to be modified. You must specify a binary transfer admission in the form x\'...\' or X\'...\' .

@a for *transfer admission*
: modifies either the FT profile specified with *profile name* (see above) or (if no profile name was specified) all the profiles that come into question.

@n for *transfer admission*
: selects all FT profiles without transfer admission.

transfer admission not specified
: causes to query the transfer admission on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the transfer admission. To exclude the possibility of typing errors, the program prompts you to enter the transfer admission a second time. If you just press <ENTER>, this has the same effect as specifying *@a*.

,user ID
: As the FTAC administrator, you can specify any login name here.

@a for *user ID*
: If you specify *@a* as the FTAC administrator, you can modify the FT profiles for any login names.

@adm for *user ID*

If you specify *@adm* as the FTAC or ADM administrator, you can modify ADM profiles (corresponding to *-ff=c*). However, you can neither change this property (*-ff=c*) nor the user ID (*-ua* option).

user ID not specified

modifies only profiles belonging to the user's own login name, regardless of who issues the command.

-s not specified

if *@a* is specified for *profile name*, all the FT profiles belonging to the login name under which the *ftmodp* command is issued are modified. Otherwise, the FT profile with the specified name is modified.

-ua=[user ID]**,**[password | **@n**]

With *-ua*, the FTAC administrator can assign any desired FT profile of a login name to another login name.

user ID

As the FTAC administrator, you can specify any login name here.

,password

specifies the password for a login name. A binary password must be specified in the form x\'...\' or X\'...\' . The FT profile for the login name is valid only so long as the password *password* is valid for the login name. When the password is changed, the profile can no longer be used (not locked!).

@n for *password*

In this case, the FTAC administrator cannot specify any transfer admission for the FT profile if you do not have FT administrator privileges. An existing transfer admission will be automatically deleted in this case.

comma only (,) no *password* specified

causes FTAC to query the password on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the transfer admission. In this case, single quotes must not be escaped by a backslash (\).

user ID only (without comma and *password*) specified

means that the profile is valid again for all passwords of the specified login name *user ID*.

-ua␣not specified

the login name of this FT profile remains unchanged.

-nn=profile name | **@s**

-nn can be used to assigns a new name to one of your FT profiles.

@s for *profile name*

Makes the admission profile the standard admission profile for the user ID. If the admission profile previously had a transfer admission, you must also specify *-tad=@n*.

-nn not specified

leaves the profile name unchanged.

-tad=[transfer admission | **@n**]

allows you to modify the transfer admission of an FT profile. As the FTAC administrator, you can also modify the transfer admissions for other login names if you have FT administrator privileges.

If the modified admission profile is a standard admission profile (*ftmodp @s* or *-nn=@s*), only *-tad=@n* is permitted.

transfer admission

The transfer admission must be unique within your Unix system so that there are no conflicts with transfer admissions defined by other FTAC users for other access permissions. A binary transfer admission must be specified in hexadecimal format in the form x\'...\' or X\'...\' . If the transfer admission you select has already been assigned, FTAC rejects the *ftmodp* command and issues the message `Transfer admission already exists.`

@n for *transfer admission*

disables the old transfer admission.

@n must be specified if you convert an admission profile that has a transfer admission to a standard admission profile using *-nn=@s*.

transfer admission not specified

-tad= causes FTAC to prompt you to enter the transfer admission after the command has been entered. Your entry is not displayed to prevent unauthorized persons from seeing the transfer admission. To exclude the possibility of typing errors, the program expects you to enter the transfer admission a second time as an entry check.

The transfer admission is not queried when a standard admission profile is changed. The following message is issued: `Transfer admission of standard profile must be @n.`

-tad not specified

does not modify the transfer admission of the FT profile.

-v=y | **-v=n**

-v defines the status of the transfer admission.

y the transfer admission is not disabled (it is valid).

n transfer admission is disabled (it is not valid).

-v is ignored if the modified profile is a standard admission profile.

-v not specified

the transfer admission status remains unchanged.

-d=[yyyymmdd]

-d specifies the period during which the transfer admission can be used. The FT profile is disabled when this period has expired.
You can specify an eight-digit date (e.g. 20170602 for June 2, 2017). The transfer admission can no longer be used after 00:00 hours on the specified day. The largest possible value that can be specified for the date is 20380119 (January 19, 2038).

yyyymmdd not specified

when *-d=* is specified, the previous setting is cancelled, i.e. the time restriction is removed from the transfer admission.

-d is ignored if the modified profile is a standard admission profile.

-d not specified

the previous time restriction defined for the transfer admission remains unchanged.

-u=pr | **-u=pu**

using *-u*, you can control how FTAC reacts when someone attempts to assign an existing transfer admission to an FT profile. Normally, the transfer admission must be disabled immediately, by designating it as private.

Transfer admissions that do not require as much protection, can be designated as public. This means that they are not disabled even when a user attempts to assign another transfer admission of the same name.
Possible values:

pr (default value)

the transfer admission is disabled as soon as someone with another login name attempts to specify a transfer admission of the same name (private). In this case, the *-u* parameter is set to *no time restriction* at the same time.

pu the transfer admission is not disabled, even if someone attempts to specify a transfer admission of the same name (public).

-u is ignored if the modified profile is a standard admission profile.

-u not specified

the previous setting remains unchanged.

-priv=y | **-priv=n**

This option is used by the FTAC administrator to grant privileged status to an FT profile.

y grants privileged status to the FT profile. The FT administrator's entries in the admission set are ignored for requests executed with a privileged FT profile, i.e., if the user uses the *-iml, -iis, -iir, -iip* or *-iif* options in the FT profile, both the user' s entries (MAX. USER LEVELS) and the administrator' s entries (MAX. ADM LEVELS) are ignored.

n withdraws the privileged status, if it had been granted, from the FT profile.

-priv not specified

does not modify the privileged status of the FT profile.

-iml=y | **-iml=n**

-iml (ignore max. level) is used to specify whether the FT profile is to be restricted by the values in the admission set. The user can override the entries he/she made himself or herself (the MAX. USER LEVELS) for requests using this FT profile. If the FT profile is also privileged by the FTAC administrator, the entries made by the FTAC administrator (the MAX. ADM LEVELS) can also be ignored. This FT profile would then allow *inbound* basic functions to be used which are disabled in the admission set.

y allows the values in the admission set to be ignored.

n restricts the functionality of the profile to the values in the admission set.

-iml not specified

causes the values specified in the profile for the basic functions to apply unchanged.

-iis=y | **-iis=n**

-iis (ignore inbound send) allows the value for the basic function *inbound send* in the admission set to be ignored (for details, see *-iml*).

y allows the basic function *inbound send* to be used even if it is disabled in the admission set. At the same time, component "display file attributes" of the basic function *inbound file management* can be used (see table at *-iif*).

Specifying this option is enough as long as the basic function *inbound send* was disabled by the user, but if it was disabled by the FTAC administrator, it is also necessary that he/she grant privileged status to the FT profile.

n restricts the profile to the value in the admission set for the basic function *inbound send*.

-iis not specified

causes the values specified in the profile for the basic function *inbound send* to apply unchanged.

-iir=y | **-iir=n**

-iir (ignore inbound receive) allows the value for the basic function *inbound receive* in the admission set to be ignored (for details, see *-iml*).

y allows the basic function *inbound receive* to be used even if it is disabled in the admission set. At the same time, subcomponents of the basic function *inbound file management* can also be used (see table at *-iif*).

Specifying this option is enough as long as the basic function *inbound receive* was disabled by the user, but if it was disabled by the FTAC administrator, it is also necessary that he/she grant privileged status to the FT profile.

n restricts the profile to the value in the admission set for the basic function *inbound receive*.

-iir not specified

causes the values specified in the profile for the basic function *inbound receive* to apply unchanged.

-iip=y | **-iip=n**

-iip (ignore inbound processing) allows the value for the basic function *inbound follow-up processing + preprocessing + postprocessing* in the admission set to be ignored (for details, see *-iml*).

y allows the basic function *inbound follow-up processing + preprocessing + postprocessing* to be used even if it is disabled in the admission set. Specifying this option is enough as long as the function was disabled by the user, but if it was disabled by the FTAC administrator, it is also necessary that he/she grant privileged status to the FT profile.

n restricts the profile to the value in the admission set for the basic function *inbound follow-up processing + preprocessing + postprocessing*.

-iip not specified

causes the values specified in the profile for the basic function *inbound follow-up processing + preprocessing + postprocessing* to apply unchanged.

-iif=y | **-iif=n**

-iif (ignore inbound file management) allows the values for the basic function *inbound file management* in the admission set to be ignored (for details, see *-iml*).

y allows the basic function *inbound file management* to be used even if it is disabled in the admission set.

Specifying this option is enough as long as the basic function *inbound file management* was disabled by the user, but if it was disabled by the FTAC administrator, it is also necessary that he/she grant privileged status to the FT profile.

n restricts the profile to the value in the admission set for the basic function *inbound file management*.

The following table shows which subcomponents of the file management can be used under which conditions.

Inbound file management function	**Values of the admission set or extension in profile**
Display file attributes	Inbound Send (IBS) enabled
Modify file attributes	Inbound Receive (IBR) **and** Inbound File Management (IBF) enabled
Rename files	Inbound Receive (IBR) **and** Inbound File Management (IBF) enabled
Delete files	Inbound Receive (IBR) enabled **and** Write mode = overwrite in profile
Display directories	Inbound File Management (IBF) enabled
Create, rename and delete directories	Inbound File Management (IBF) enabled **and** direction = from partner in profile

iif not specified
causes the values specified in the profile for the basic function *inbound file management* to apply unchanged.

-ff=[**t**][**m**][**p**][**r**][**a**][**l**] | **-ff=c**
-ff defines the FT function for which the FT profile can be used. With the exception of *c*, these letters can be combined in any way (*tm, mt, mr,* ...). *c* must not be combined with other values. Please observe the note concerning the description of *-ff=c* on page 257.

t (transfer) The FT profile can be used for the file transfer functions "Transfer files", "Display file attributes", and "Delete files".

m (modify file attributes) The FT profile can be used for the file transfer functions "Display file attributes" and "Modify file attributes".

p (processing) The FT profile can be used for the file transfer functions "File Preprocessing" or "File Postprocessing". The FT function "Transfer files" must also be permitted.

Specification of *p* has no significance for profiles with a file name prefix (*-fnp=*) or a file name (*-fn=*) since, in this case, the first character of the file name or file name prefix decides whether the profile can only be used for preprocessing and postprocessing ("|") or only for file transfer/file management (no "|").

The use of follow-up processing is not controlled by *-ff=*, but by *-lf=* and *-ls=*.

r (read directory) The FT profile can be used for the file transfer functions "Display directories" and "Display file attributes".

a (administration)
The admission profile is allowed to be used for the "remote administration" function. This means that it authorizes a remote administration server to access the local openFT instance. To do this, the associated transfer admission must be configured in the remote administration server.
-ff=a may only be specified by the FT administrator or FTAC administrator.

l (logging)
The admission profile is allowed to be used for the "Receive ADM traps" function. This allows another openFT instance to send its ADM traps to the remote administration server via this profile. This specification only makes sense if the local openFT instance is flagged as a remote administration server (*ftmodo -admcs=y* command).
-ff=l may only be specified by the FT administrator.

c (client access)
The admission profile is allowed to be used for the "access to remote administration server" function (ADM profile). This allows a remote administrator on a remote computer to use this profile to access the local remote administration server and issue remote administration requests. The local openFT instance must be flagged as a remote administration server (*ftmodo -admcs=y* command).

ff=c may only be specified by the ADM administrator.

The value *c* must not be combined with any other value. In addition, an FT profile created with *-ff=c* cannot be changed into a FT profile using the other FT functions (*t, m, p, r, a* or *l*) and vice versa.

No function specified
Specifying *-ff=* allows you to undo any specification with regard to the functions. All file transfer functions are then permitted (corresponds to *tmpr*), but not the remote administration functions (*a, c*) and ADM trap functions (*l*).

-ff not specified
The previous specification with respect to the functions remains unchanged.

-dir=f | **-dir=t** | **-dir=ft**
specifies for which transfer direction(s) the FT profile may be used. Possible values for the direction: *f*, *t*, *ft*, *tf*.

f allows data transfer only from a partner system to the local system.

t allows data transfer only from the local system to the remote system. It is thus not possible to create, rename or delete directories.

ft, tf
transfer direction is not restricted in the profile.

-dir not specified
leaves the transfer direction entries in the FT profile unchanged.

-pn=[partner1[**,**partner2, ...]]
You use *-pn* to specify that this admission profile is to be used only for FT requests which are processed by a certain partner system. You can specify the name of the partner system in the partner list or the address of the partner system. For details on address specifications, see section “Specifying partner addresses” on page 68.

You can specify more than one partner system (maximum 50) with a maximum total of 1000 characters.

partner1[,*partner2, ...*] not specified
-pn= cancels a previous restriction defined for partner systems so that the FT profile can be used by every partner system.

-pna=partner1[**,**partner2, ...]
-pna adds one or more partner system(s) to the list of permitted partner systems. Up to 50 partner systems can be entered in the list (max. 1000 characters).

If the list has been empty up to now, then the profile is limited to the specified partner system(s).

-pnr=partner1[**,**partner2, ...]
-pnr deletes one or more partner system(s) from the list of permitted partner systems.

Please note: As soon as you delete the last partner remaining in the list, the profile can be used by every partner system.

-pn, *-pna* and *-pnr* not specified
causes the entries for permitted partner systems to apply unchanged.

-fn=[file name]

-fn specifies which file(s) under your login name may be accessed using this FT profile. If you specify a fully qualified file name, only the file with this name can be transferred.
If the file name ends with %unique or %UNIQUE, this string is replaced by a string which changes for each new call on file transfer or file management requests. In Unix systems, this string is 14 characters long. In addition, a suffix separated by a dot may be specified after %unique or %UNIQUE, e.g. `file1%unique.txt`. Only the already converted file name is displayed in both the log and the messages.
If *file name* starts with a "|" (pipe character) then it is interpreted as a preprocessing or postprocessing command.

file name not specified

-fn= allows you to cancel a file name entry. This also applies to a prefix assigned with *-fnp*. The FT profile then permits unrestricted access to all files.

-fn not specified

leaves the file name entries in the FT profile unchanged.

-fnp=file name prefix

restricts access to a set of files whose names begin with the same prefix. FTAC adds the character string specified as *file name prefix* to the file name in the request and attempts to transfer the file with the expanded name.
For example, if this option is specified as *-fnp=scrooge/* and the request contains the file name *stock*, the file is transferred as *scrooge/stock*.

In this way, you can designate the files you have released for openFT. If the *-fnp* option was used to specify a prefix, the file name specified in the request must not contain the character string *../* to avoid (unintentionally) changing directories. You should also ensure that there is no chance for a symbolic link to cause a jump to another place in the file tree.
%unique or %UNIQUE cannot be used for a file name prefix. In the case of a file transfer or file management request, the user can use a file name ending with %UNIQUE (or %UNIQUE.*suffix* or %unique or %unique.*suffix*) to generate a unique file name with the prefix specified here.

A file name prefix which starts with the | character indicates that the FTAC profile can only be used for file transfer with preprocessing and postprocessing, since the file name created using the prefix and the name specified for the ncopy or *ft* command also starts with the | character. In this case, no follow-up commands may be specified.

On Unix systems, the shell metacharacters | ; & < > and "newline" may only be specified if they are enclosed in '...' (single quotes) or "..." (double quotes) or if each of them is escaped with "\" (backslash). The character `

(accent grave) and the string $((dollar+open bracket) may only be specified if they are enclosed in '...' (single quotes) or if they are specified directly after a backslash ("\").

The following strings may not be specified in the command that uses the profile
– .. (two dots)
– .\ (dot + backslash)
– .' (dot + single quote)
This makes it impossible to navigate to higher-level directories.

file name prefix can be up to 511 bytes in length.

-fn= allows you to cancel a file name prefix entry, see above.

Special cases

- You must specify a file name or file name prefix which starts with the string "|ftexecsv␣" for FTAC profiles which are to be used exclusively for the *ftexec* command. If a command prefix is also to be defined, you must specify it as follows:

  -fnp="|ftexecsv␣-p=*command prefix***"**
  (e.g.: `-fnp="|ftexecsv␣-p=\"ftshwr␣\""`)

  The same restrictions apply to the command string of the *ftexec* call as to the filename prefix during preprocessing and postprocessing.

- For FTAC profiles that are only to be used for getting monitoring data, specify the filename prefix "|*FTMONITOR ". The functions of the profile must permit File Preprocessing (*-ff=tp*). For details, see the *ftcrep* command, Example 3 on page 201.

-fnp not specified
leaves the *file name prefix* entries in the FT profile unchanged.

-ls= | **-ls=@n** | **-ls=**command1
specifies follow-up processing which is to be performed under your login name in the event that **file transfer** is **successful**. If *-ls* is specified, no success follow-up processing may be requested in the file transfer request. Specifying *-ls* only makes sense if you also make an entry for *-lf* (see below) to preclude the possibility that an intentionally unsuccessful request can circumvent the *-ls* entry. If you have defined a prefix for the file name with *-fnp* and plan follow-up processing for this file, you must specify the complete file name here.

@n for *command1*
If you enter *-ls=@n*, no follow-up processing is then permitted in the FT profile in the event that file transfer is successful.

command1 not specified

-ls= allows you to cancel a follow-up-processing entry. The FT profile then no longer restricts success follow-up processing in the local system. This is also a way to cancel a prefix for the follow-up processing defined with *-lsp*.

-ls not specified

leaves the entries in the FT profile for follow-up processing in the event that file transfer is successful unchanged.

-lsp=[command2]

-lsp defines a prefix for follow-up processing in the local system in the event that **file transfer** is **successful**. FTAC then adds the character string *command2* to the follow-up processing specified in the FT request and attempts to execute the resulting command. For example, if this option is specified as *-lsp='lpr␣'* and the request specifies *file1.txt* as follow-up processing, FTAC executes *lpr␣file1.txt* as follow-up processing.
Prefix, suffix and follow-up processing commands must together not be longer than 1000 bytes (for the representation in UTF-8, see page 163).

Please also bear in mind the information provided on the *-ls* option!

If a prefix was defined with *-lsp*, the character set available for specifying follow-up processing in the FT request is restricted to:

– alphanumeric characters (letters and digits)
– the special characters + = / ! ␣ - , @ _ " $ '
– a period (.) between alphanumeric characters

You can cancel an existing prefix by specifying *-ls*=.

command2 not specified

-lsp= cancels the entry in the FT profile for a follow-up processing prefix after successful file transfer.

-lsp not specified

leaves the prefix entries in the FT profiles for follow-up processing in the event that file transfer is successful unchanged.

-lss=[command3]

-lss defines a suffix for follow-up processing in the local system in the event that **file transfer** is **successful**. FTAC then appends the character string *command3* to the follow-up processing specified in the FT request and attempts to execute the resulting command. For example, if this option is specified as *-lss=␣file2.txt* and the request specifies *lpr* as follow-up processing, FTAC executes *lpr␣file2.txt* as follow-up processing.
Prefix, suffix and follow-up processing commands must together not be longer than 1000 bytes (for the representation in UTF-8, see page 163).
Please also bear in mind the information provided on the *-ls* option!

If a suffix was defined with *-lss*, the character set available for specifying follow-up processing in the FT request is restricted to:
– alphanumeric characters (letters and digits)
– the special characters + = / ! ␣ - , @ _ " $ '
– a period (.) between alphanumeric characters

command3 not specified
-lss= cancels the entry in the FT profile for a follow-up processing suffix after successful file transfer.

-lss not specified
leaves the suffix entries in the FT profiles for follow-up processing in the event that file transfer is successful unchanged.

-lf= | **-lf=@n** | **-lf=**command4
-lf specifies follow-up processing to be executed under your login name if the **file transfer** is **aborted** due to an error. If *-lf* is specified, no failure follow-up processing may be requested in the FT request. Making an *-lf* entry only makes sense if you also make an entry for *-ls* (see above) to preclude the possibility that a successful request can circumvent the *-lf* entry. If you have defined a prefix for the file name with *-fnp* and plan follow-up processing for this file, you must specify the complete file name here.

@n for *command4*
-lf=@n is specified, no follow-up processing is then permitted in the FT profile in the event of an unsuccessful file transfer.

command4 not specified (*-lf=*)
-lf= allows you to cancel an entry for follow-up-processing in the event that file transfer is unsuccessful. The FT profile then no longer restricts failure follow-up processing in the local system. This is also a way to cancel a prefix defined with *-lfp*.

-lf not specified
leaves the entries in the FT profiles for failure follow-up processing after unsuccessful file transfer unchanged.

-lfp=[command5]
defines a prefix for follow-up processing in the local system in the event that **file transfer** is **unsuccessful**. FTAC then adds the character string *command5* to the follow-up processing specified in the FT request and attempts to execute the resulting command. For example, if this option is specified as *-lfp='lpr␣'* and the request specifies *error.txt* as follow-up processing, FTAC executes *lpr␣error.txt* as follow-up processing.
Prefix, suffix and follow-up processing commands must together not be longer than 1000 bytes (for the representation in UTF-8, see page 163).

Please also bear in mind the information provided on the *-lf* option!

If a prefix was defined with *-lfp*, the character set available for specifying follow-up processing in the FT request is restricted to:
– alphanumeric characters (letters and digits)
– the special characters + = / ! ␣ - , @ _ " $ '
– a period (.) between alphanumeric characters

You can cancel an existing prefix by specifying *-lf*=.

command5 not specified
-lfp= cancels the follow-up processing prefix in the FT profile in the event of unsuccessful file transfer.

-lfp not specified
leaves the prefix entries in the FT profiles for follow-up processing in the event of unsuccessful file transfer unchanged.

-lfs=[command6]
-lfs defines a suffix for follow-up processing in the local system in the event that **file transfer** is **unsuccessful**. FTAC then appends the character string *command6* to the follow-up processing specified in the FT request and attempts to execute the resulting command. For example, if this option is specified as *-lfs=␣error.txt* and the request specifies *lpr* as follow-up processing, FTAC executes *lpr␣error.txt* as follow-up processing.
Prefix, suffix and follow-up processing commands must together not be longer than 1000 bytes (for the representation in UTF-8, see page 163).

Please also bear in mind the information provided on the *-lf* option!

If a suffix was defined with *-lfs*, the character set available for specifying follow-up processing in the FT request is restricted to:
– alphanumeric characters (letters and digits)
– the special characters + = / ! ␣ - , @ _ " $ '
– a period (.) between alphanumeric characters

command6 not specified
-lfs= cancels the follow-up processing suffix in the FT profile in the event of unsuccessful file transfer.

-lfs not specified
leaves the suffix entries in the FT profile for a follow-up processing in the event of unsuccessful file transfer unchanged.

-wm=o | **-wm=n** | **-wm=e** | **-wm=one**

-wm specifies which write modes may be used in the file transfer request and what they effect.

o (overwrite) In the FT request of openFT or FTAM partners, only *-o* or *-e* may be entered for write mode. The receive file is overwritten if it already exists, and is created if it does not yet exist.

With FTP partners, *-n* may also be entered if the file does not yet exist.

n (no overwrite) In the FT request *-o*, *-n* or *-e* may be entered for write mode. The receive file is created if it does not yet exist. If the receive file already exists, the request is not executed.

e (extend) In the FT request only *-e* may be entered for write mode, i.e. the receive file is extended by appending the transferred file to the end if the receive already exists. The receive file is created if it does not yet exist.

one means that the FT profile does not restrict the write mode.

-wm not specified

leaves the write-mode entries in the FT profile unchanged.

-c= | **-c=y** | **-c=n**

Using *-c*, you can determine whether data encryption is required or forbidden. If the setting in the profile does not correspond to the setting in the request, the request is denied. The setting is not valid for file management requests, since there is no data encryption for these requests.

y Only requests **with** data encryption may be processed using this profile.

n Only requests **without** data encryption may be processed using this profile.

neither *y* nor *n* specified

-c= resets the current setting. Requests with and without data encryption are both accepted.

-c not specified

The encryption option remains unchanged.

-txt=text | **-txt=**

-txt allows you to enter a new comment in the FT profile (up to 100 characters).

text not specified

-txt= deletes an existing comment.

-txt not specified

an existing comment remains unchanged.

As soon as you modify an admission profile, the timestamp is also updated. The timestamp is output with *ftshwp -l* (LAST-MODIF). The timestamp is also updated if you do not change the properties of the profile, i.e. if you enter *ftmodp* without any parameters.

CAUTION!

If you use the *-ff=p*, *-fn*, *-fnp*, *-ls*, *-lsp*, *-lss*, *-lf*, *-lfp* or *-lfs* options, you must remember

- that a file name restriction can be bypassed by renaming the file unless follow-up processing is also restricted;
- that follow-up processing must always be restricted for both successful and unsuccessful file transfer and, if necessary, equivalent restrictions must exist for any permitted preprocessing;
- that prefixes for the file names and follow-up processing must be matched to one another;
- that no symbolic links should occur in the part of your file tree that is referenced by the file name prefix;
- that restrictions applied to preprocessing or follow-up processing can be circumvented if it is possible to replace this command with, for example, a "Trojan horse".

Example

The transfer admission in the *goldmrep* FT profile created in the section “Examples” on page 200, is to be changed to *forScrooge*. The transfer direction is no longer to be restricted. The profile is to be used to transfer any files with the prefix *mine/*. Follow-up processing is to be prohibited entirely.

The following command has to be entered:

```
ftmodp␣goldmrep␣-tad=forScrooge␣-dir=tf\
␣-fnp=mine/␣-ls=@n␣-lf=@n
```

6.27 ftmodptn - Modify partner properties

You use the *ftmodptn* command to modify the properties of partner systems in the local system's partner list.

Please note that if you modify the partner address, it is no longer possible to convert an openFT partner into an FTP partner or FTAM partner or vice versa.

You can remove an entered dynamic partner from the partner list by setting all the properties to the default values for free dynamic partners by means of the *ftmodptn* command. The default values are the same as the default values in the *ftaddptn* command with the exception of the security level setting (option *-sl*) which must be set to *-sl=p*.

Similarly, you can add a free dynamic partner to the list by setting at least one of its attributes to a value other than the default. This is possible if *partner* does not reference a partner list entry and *-pa* is not specified.

If a partner name for which there is as yet no partner list entry is specified for *partner* and *-pa* is also specified then a new named entry is created in the partner list. This function is intended for the re-import of exported partner entries. To explicitly create new partner entries, you should use *ftaddptn*.

Format

```
ftmodptn  -h |
          <partner 1..200> | @a
        [ -pa=<partner address 1..200> ]
        [ -id=<identification 1..64> | -id= ]
        [ -ri=<routing info 1..8> | -ri=@i | -ri= ]
        [ -ptc=i | -ptc=a | -ptc= ]
        [ -pri=l | -pri=n | -pri=h ]
        [ -sl=1..100 | -sl=p | -sl= ]
        [ -st=a | -st=d | -st=ad ]
        [ -ist=a | -ist=d ]
        [ -am=n | -am=y ]
        [ -rqp=p | -rqp=s ]
        [ -tr=n | -tr=f | -tr= ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

partner | **@a**

partner is the name of the partner system in the partner list or the address of the partner system whose properties you want to modify.

@a for *partner*

Partner is not a selection criterion, i.e. you modify the properties of all the partner systems present in the partner list. This specification is only possible in combination with the options *-ptc*, *-sl*, *-st*, *-ist*, *-am*, *-rqp* and *-tr*..

Particular care is necessary when using *@a* in combination with *-sl* (security level)!

-pa=partner address

You use *-pa* to enter the address of the partner system in the following form:

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

For details concerning address specifications, see section “Specifying partner addresses” on page 68.

-pa not specified

The partner address is unchanged.

-id=identification | **-id=**

Identification unique in the network of the openFT instance in the partner system. In the case of FTAM partners, it is possible to specify an Application Entity Title in the form n1.n2.n3.n4..mmm as the identification. n1, n2 etc. are positive integer values which describe the "Application Process Title". n1 can only have the values 0, 1 or 2, n2 is restricted to values between 0 and 39 if n1 does not have the value 2. The optional Application Entity Qualifier mmm must be separated from the values of the Application Process Title by two periods. For details, see the openFT User Guide.

In the case of FTP partners, *-id* must not be specified!

identification not specified

Specifying *-id=* with no other specification sets the identification to *host* (host name) for partner entries with openFT and FTADM protocol. For FTAM partners, the identification is deleted if *-id=* is entered.

-id not specified

The setting for identification is unchanged.

-ri=routing info | **-ri=@i** | **-ri=**

If the partner system can only be accessed via an intermediate instance then you specify the address information to be used for routing by the intermediate instance in *routing info*.

@i for *routing info*

The instance identification specified in *-id=* is used as the routing information.

neither *@i* nor *routing info* specified

The specification of *-ri=* (without parameters) means that the partner system can be accessed directly, i.e. without an intermediate instance.

-ri not specified

The setting for the routing information is unchanged.

-ptc=i | **-ptc=a** | **-ptc=**

You can use *-ptc* to modify the operating parameter setting for sender verification on a partner-specific basis. These settings only affect partners which are connected via the openFT protocol and do not operate with authentication (e.g. partners with openFT V8.0 or earlier).

i (identification)

Deactivates checking of the transport address. Only the partner's identification is checked. The partner's transport address is also not checked even if extended sender verification is globally active (see the *ftmodo* command on page 229).

a (address)

Activates checking of the transport address. The partner's transport address is checked even if checking of the transport address is globally deactivated (see *ftmodo* command on page 229).

If the transport address under which the partner logs on is not the same as the entry in the partner list then the request is rejected.

neither *i* nor *a* specified

-ptc= (without parameters) means that the operating system parameters apply to sender verification.

-ptc not specified

The setting for sender verification is unchanged.

-sl=1..100 | **-sl=p** | **-sl=**

You use this option to assign a security level to the specified partner system or to all the partner systems.

A low security level means that the need for protection vis a vis this partner is low, for instance because the partner's identity has been authenticated using cryptographic methods, which means that you can be certain that the partner is genuinely who they claim to be.

A high security level means that the need for protection vis a vis this partner is high, because the identity of the partner has only been determined on the basis of their address, for instance, and that no authentication has been performed using cryptographic methods.

1..100
: Assigns a fixed security level to the partner. 1 is the lowest and 100 the highest security level.

  All integers 1 through 100 are permitted.

p
: Assigns a security level to the partner depending on the partner's attributes, i.e.:

  – Security level 10 if the partner has been authenticated.

  – Security level 90 if the partner is known in the transport system and is identified by the name it is known by in the transport system.

  – Security level 100 if the partner has only been identified by its address.

security level not specified
: *-sl=* (without parameters) means that the operating parameter setting for the security level applies (see command *ftmodo* on page 229)

-sl not specified
: The setting for the security level is unchanged.

-pri=l | **-pri=n** | **-pri=h**

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

l (low)
: The partner is assigned a low priority.

n (normal)
: The partner is assigned a normal priority.

h (high)
: The partner is assigned a high priority.

-pri not specified
: The priority setting remains unchanged.

-st=a | **-st=d** | **-st=ad**
: This option allows you to control how locally submitted asynchronous file transfer requests to the specified partner system or systems are processed.

  a (active)
  : Locally submitted asynchronous file transfer requests are processed if the asynchronous openFT server is started.

  d (deactivated)
  : Locally submitted asynchronous file transfer requests are initially not processed but are stored in the request queue.

  ad (automatic deactivation)
  : Multiple consecutive unsuccessful attempts to establish a connection to this partner system result in its deactivation. If you want to perform file transfer again with this system, you must explicitly activate it with *ftmodptn -st=a*.

    The maximum number of such unsuccessful attempts is 5. After a connection has been established successfully, the counter is reset to 0.

-st not specified
: The processing mode is unchanged.

-ist=a | **-ist=d**
: This option allows you to control how file transfer requests issued remotely by the specified partner system or partner systems are processed.

  a (active)
  : File transfer requests issued remotely are processed if the asynchronous openFT server is started.

  d (deactivated)
  : Synchronous file transfer requests issued remotely are rejected. Asynchronous file transfer requests issued remotely by this partner are stored there and cannot be processed until this partner is activated again with *-ist=a*.

-ist not specified
: The processing mode is unchanged.

-am=n | **-am=y**
: You can use *-am* (authentication mode) to force partner authentication.

  n Authentication is not forced, i.e. this partner is not restricted with regard to authentication.

  y Authentication is forced, i.e. requests are only processed if the local system is successfully able to authenticate the partner, see page 77.

-am not specified
: The authentication mode is unchanged.

-rqp=p | **-rqp=s**
: You use this option (rqp = request processing) to control whether asynchronous outbound requests to this partner are always run serially or whether parallel requests are permitted.

  p (parallel)
  : Parallel connections to this partner are permitted.

  s (serial)
  : Parallel connections to this partner are not permitted. If multiple file transfer requests to this partner are pending then they are processed serially. A follow-up request is not started until the preceding request has terminated.

-rqp not specified
: The operating mode is unchanged.

-tr=n | **-tr=f** | **-tr=**
: You can use this option to modify the operating parameter settings for the partner selection for the openFT trace function on a partner-specific basis.

  n (on)
  : The trace function is active for this partner or for all the partners. However, a trace is only written if the openFT trace function has been activated via the operating parameters. In this case, this setting for *ftmodptn* takes priority over the partner selection for the trace function in the operating parameters. See page 229ff, *ftmodo*, *-tr* option.

  f (off)
  : The trace function is deactivated for this partner or for all partners.

  neither *n* nor *f* specified
  : *-tr=* (without parameters) means that the operating parameter setting for the partner selection in the openFT trace function applies (see the *ftmodo* command on page 229).

-tr not specified
: The setting for the trace function is unchanged.

6.28 ftmodr - Change the property of requests

With the *ftmodr* command, you can change the priority of requests you have issued, or of a group of requests, for example all the requests to a particular partner. Furthermore, you have the option of changing the order of requests within a priority.

As the FT administrator, you can change the priority of all requests in the system.

Format

ftmodr -h |
[-ua=<user ID 1..32> | -ua=@a]
[-pn=<partner 1..200>]
[-fn=<file name 1..512>]
[-pr=n | -pr=l][-qp=f | -qp=l]
[<request ID 1..2147483647>]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-ua=user ID | **-ua=@a**
You use *-ua* to specify the user ID for which requests are to be modified.

user ID
As FT administrator, you may specify any user ID here.

@a As FT administrator, you can specify *@a* to modify requests relating to all user IDs.

-ua= not specified
Your own user ID is the selection criterion. Exception: you called the command as FT administrator and also specified a request ID: in this case, the presetting is *@a*.

-pn=partner
You use *-pn* to specify a name or an address for the partner system for which you want to modify requests. The partner should be specified in the same way as in the request or as it is output in the *ftshwr* command without the option *-s*, *-l* or *-csv*. If openFT finds a partner in the partner list that corresponds to the specified partner address then *ftshwr* indicates the name of the partner even if a partner address was specified on request entry.

-fn=file name
: You use *-fn* to specify the file name for which requests are to be modified. Requests which access this file in the local system are modified.

  You must specify the file name that was used when the request was created. This file name is also output by the *ftshwr* command without the *-fn* option.

  Wildcards may not be used in the file name.

-pr=n | **-pr=l**
: indicates the new priority. The following values are possible:

  n (normal)
  : the request has the priority "normal".

  l (low)
  : the request has the priority "low".

-qp=f | **-qp=l**
: indicates the position of the request within the same priority. The following values are possible:

  f (first)
  : the request is placed at the top of the list of requests with the same priority.

  l (last)
  : the request is placed at the bottom of the list of requests with the same priority.

request ID
: *request ID* is used to specify the identification of a specific request that is to be modified. The request ID is output on the screen when reception of the request is confirmed. It can also be displayed using the *ftshwr* command.

If you have specified a request ID but the other specified selection criteria do not match the request then the request is not modified and the following error message is output:

```
ftmodr: Request request ID not found
```

6.29 ftmonitor - Call the openFT Monitor for displaying measurement data

The *ftmonitor* command calls the openFT Monitor in which the monitoring data collected during openFT operation is displayed. openFT can be running on the local system or on a remote system. The openFT Monitor can only be called if monitoring has been explicitly activated by the administrator on the relevant system (e.g. using the *ftmodo -mon=n* command) and the asynchronous openFT has been started.

Note that you require a graphics-capable terminal to use the *ftmonitor* command.

Format

ftmonitor -h |
[-lay=<monitor layout file name 1..512>]
[-po=<polling interval 1..600>]
[<partner 1..200> [
<transfer admission 8..67> |
<user ID 1..67>[,[<account 1..64>][,[<password 1..64>]]]]]

Description

-h Outputs the command syntax. Any specifications after *-h* are ignored.

-lay=monitor layout file name
Name of the Monitor layout file. This file describes what monitoring data is output and how it is presented.

The name of the layout file must be specified with the suffix *.ftmc*. This suffix is automatically assigned by the monitor when the file is saved if it was not explicitly specified there.

The content of the layout file is also generated by the Monitor. You must not change the content of the layout file.

After the default Monitor window has been opened for the first time (without specifying *-lay*), you can create and save your own layout file. To do this, choose a different layout from the *View* menu of the Monitor window, for instance, or set a different value using the selection icon on the top right and store the setting under a name of your choice. Refer to the online Help system of the openFT Monitor window for details.

-lay not specified
If you do not specify *-lay*, the default Monitor window is opened. This contains a chart showing the monitoring value *Networkb/sec of all Requests* (corresponds to the parameter *ThNetbTtl* in the command *ftshwm*).

-po=polling interval
: Polling interval in seconds.

  Possible values: 1 through 600.
  Default value: 1

partner
: Name or address of the partner system for which monitoring data is to be shown. The partner must be an openFT partner (i.e. communication via the openFT protocol) and must support the collection of monitoring data, i.e. the openFT version of the partner must be at least V11.

  In addition, the partner's asynchronous openFT server must be started and monitoring must be activated in its operating parameters.

partner not specified
: If you do not specify a partner, the monitoring data of the openFT instance on the local computer is output.

transfer admission | user ID[**,**[account][**,**[password]]]
: Transfer admission for the partner system. File transfer and preprocessing/postprocessing must be permitted under the specified transfer admission.

  You can specify this transfer admission

  – as an FTAC transfer admission if FTAC is used in the remote system or destination instance. For this purpose, a special admission profile with the filename prefix "|*FTMONITOR " can be set up on the partner system that only permits monitoring data to be collected. You will find an example under *ftcrep* on page 201.

  – or as a login/LOGON admission using the syntax of the remote system (*user ID*, where necessary with *account* and/or *password*).

transfer admission not specified
: If you do not specify a transfer admission for a remote partner system, the system prompts you for it in a dialog box. The entry made for the password or the FTAC transfer admission remains invisible. Asterisks (*****) are displayed as replacement characters.

Messages from the openFT Monitor

The openFT Monitor issues error messages in the form of a dialog box. It terminates automatically if an error occurs or if monitoring is terminated in the system being monitored.

If the layout of the Monitor window is changed and if openFT is terminated before the changed layout is saved, the openFT Monitor issues a message and queries whether the layout is to be saved.

6.30 ftremptn - Remove a partner from the partner list

ftremptn removes a partner form the partner list.

Format

ftremptn [-h] |
<partner 1..200>

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

partner
Specifies the partner that is to be removed from the partner list. You can specify the name in the partner list or the partner's address. The name and address are displayed using the *ftshwptn* command.

All requests stored for this partner in the request queue are deleted. This is even the case for requests with a status which means that they are known to the partner system. Since this can lead to inconsistencies, you should only remove a partner from the partner list if either there are no more requests for this partner in the request queue or if you can be sure that the partner system will not become active again.

6.31 ftsetjava - Manage link to the Java executable

ftsetjava is used to set a link to the Java executable.

ftsetjava is used implicitly during installation of openFT. In addition, you can also call *ftsetjava* as administrator in order to

- see what file is referenced by the link to the Java executable used by openFT.
- set the link if Java was not installed or if an incorrect version was installed at the time when openFT was installed or if the installation path of the Java executable has changed.
- see what Java installations are present in the directories searched by openFT.

Format

ftsetjava [@s | @a | <file name 1..512>]

Description

@s Sets the link to the Java executable.
If the attempt to set a link to the Java executable fails because no suitable Java installation is available, an appropriate message is output to *stdout*. A warning is also issued if this happens during installation of openFT.

@a Shows all the Java executables installed in the search path. Any subsequent call to *ftsetjava @s* is successful if and only if at least one of these installations meets the requirements of openFT with respect to the version. The file whose version is closest to that of the required Java version 1.5 is then used as the source of the link. If multiple Java executables with the same version are installed then the first of these displayed in the list is used.

file name
Sets the link to the specified Java executable.

You must specify the fully qualified filename of a Java executable that matches the version requirements stipulated by openFT. If the attempt to set a link to the Java executable fails, a message to this effect is issued to standard output.

neither *@s* nor *@a* nor *file name* specified
If *ftsetjava* is called without parameters, it outputs the complete path of the executable used by openFT.

6.32 ftshwa - Display admission sets

ftshwa stands for "show admission set", and allows you to examine admission sets.

As the FTAC administrator, you can obtain information on all admission sets in your system.

As the FT administrator, you can determine the FTAC administrator and the ADM administrator.

It outputs the following information:

- what limit values the owner of the user ID has set for the individual basic functions
- what limit values the FTAC administrator has set for the user ID for the individual basic functions,
- whether or not the admission set has the FTAC privilege (i.e. if the owner of the admission set is the FTAC administrator).
- whether or not the admission set has the ADM privilege (i.e. if the owner of the admission set is the ADM administrator).

Format

ftshwa -h |
[<user ID 1..32> | @a | @s][-csv]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

user ID | **@a** | **@s**
specifies the user ID for which the admission set is to be displayed.

user ID
As the FTAC administrator, you can specify any login name desired.

If a login name longer than 8 characters is specified, the first 7 characters are output followed by an asterisk (*).

@a for *user ID*
When entered by the FTAC administrator, *@a* displays information on the standard admission set and all admission sets that differ from it.

When entered by the FT administrator (who is not the FTAC administrator), *@a* displays information on the own admission set, the standard admission set and the admission set of the FTAC administrator.

@s for *user ID*
returns information only on the standard admission set.

If you specify a non-existent login name, the current standard admission set is displayed for this login name.

user ID not specified
FTAC displays information on the admission set of the login name under which *ftshwa* was entered.

-csv Specifying *-csv* indicates that the FT admission sets are to be output in the CSV format. The values in the output are separated by semicolons.

-csv not specified
The FT admission sets are output in the standard format.

6.32.1 Output format of ftshwa

Example for outputting all admission sets:

```
ftshwa @a
                MAX. USER LEVELS              MAX. ADM LEVELS         ATTR
 USER-ID  OBS  OBR  IBS  IBR  IBP  IBF  OBS  OBR  IBS  IBR  IBP  IBF
 *STD     100  100  100  100  100  100  100  100  100  100  100  100
 root      50   50    1    1    1    1  100* 100* 100* 100* 100* 100* PRIV,ADMPR
 smith     90   90    0    0    0   90  100* 100* 100* 100* 100* 100*
```

Explanation

USER-ID
The USER-ID column contains the login names to which the respective admission sets belong. If a login name longer than 8 characters is specified, the first 7 characters are output followed by an asterisk (*).

MAX. USER LEVELS / MAX. ADM LEVELS
The six columns under MAX. USER LEVELS show the values specified by each of these FTAC users for their respective admission sets. The six columns under MAX. ADM LEVELS contain the values set by the FTAC administrator.
The lower of the two values determines whether or not the owner of this admission set may use the basic function specified.

The names of the basic functions are abbreviated as follows:
OBS = **O**UT**B**OUND-**S**END
OBR = **O**UT**B**OUND-**R**ECEIVE
IBS = **I**N**B**OUND-**S**END
IBR = **I**N**B**OUND-**R**ECEIVE
IBP = **I**N**B**OUND-**P**ROCESSING
IBF = **I**N**B**OUND-**F**ILE-MANAGEMENT

The values in the admission set have the following meaning:

0	The basic function is disabled.
1..99	The basic function is only released for partner systems with the same or a lower security level. You can use the *ftshwptn* command to display a partner system's security level.
100	The inbound basic function is enabled for all partner systems.

An asterisk '*' after the value indicates that this entry was taken from the standard admission set and will automatically be modified if the value in the standard admission set is changed.

ATTR PRIV in the ATTR column indicates the privileged admission set. *root* is the FTAC administrator in this example.

ADMPR in the ATTR column indicates the ADM administrator. This means that *root* is also the administrator of the remote administration server.

6.33 ftshwatp - Display ADM traps

If you are the FT administrator of the ADM trap server, *ftshwatp* allows you to obtain information on the ADM traps sent to the ADM trap server and stored in the ADM trap log file there.

If the ADM trap server is also used as remote administration server, both the ADM administrator and the remote administrators can view traps.

- If you are the ADM administrator of the remote administration server, you can view all ADM traps.
- If you are a remote administrator, you can view "your" ADM traps (locally or with *ftadm*). This means that you only see the ADM traps of those openFT instances for which you have at least FTOP permission. See the section “ftshwc - Show openFT instances that can be remotely administered” on page 288.

The ADM traps are identified by trap IDs. The trap IDs are assigned in ascending sequence. For technical reasons, the numbering sequence is not always unbroken. If no other specifications are made, openFT always outputs the most recent ADM trap. When requested, openFT outputs all the ADM traps up to the number specified in the command.

The ADM traps are stored in the ADM trap log file. The maximum number of stored ADM traps depends on the maximum possible size of the ADM trap log file. If the maximum number of ADM traps is exceeded, the records with the lowest trap ID are overwritten by the current records. For further details, see page 147.

You can choose between three output formats, short output format, detailed output format and CSV output format (**C**haracter **S**eparated **V**alue).

The ADM traps are output to standard output.

Format

```
ftshwatp -h |
        [ -rg=[[[[yyyy]mm]dd]hhmm |
                  #1..999999999999999999 ][-
                [[[[yyyy]mm]dd]hhmm |
                [ #1..999999999999999999 ] ]
        [ -src=<partner 1..200> ]
        [ -tt=[fts][,][pts][,][ptu][,][rqc][,][rqf][,][rqs] ]
        [ -nb=1.. 9999999 | -nb=@a ]
        [ -l | -csv ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-rg=[[[[yyyy]mm]dd]hhmm][**-**[[[[yyyy]mm]dd]hhmm]
With *-rg*, you can optionally specify the start or end of a time period.

[[[yyyy]mm]dd]hhmm
If you specify a time as 4 digits, this is interpreted as hours and minutes. 6 digits are interpreted as day (date) and time in hours and minutes, 8 digits as month, day and time in hours and minutes and 12 digits as year, month, day and time in hours and minutes. The largest possible value that can be entered for the date is 20380119 (19th January 2038).
openFT then outputs the ADM traps that lie between the specified limits.

-rg=[[[yyyy]mm]dd]hhmm
The ADM traps that occurred at the specified time are output.

-rg=[[[yyyy]mm]dd]hhmm**-**[[[yyyy]mm]dd]hhmm
The time period begins with the start time and ends with the second time specified.

If a number is specified with *-nb* that is smaller than the number of ADM traps in the period, the required number of ADM traps up to the end time is output.

-rg=[[[yyyy]mm]dd]hhmm**-**
The time period begins at the start time and ends with the most recent ADM trap entry.

If a number is specified with *-nb* that is smaller than the number of ADM traps in the period, the most recent ADM traps are output.

-rg=-[[[yyyy]mm]dd]hhmm
The time period ends at the specified time.

If a number is specified with *-nb* that is smaller than the number of ADM traps in the period, the required number of ADM traps up to the end time is output.

-rg=[**#**1..99999999999999999][**-**[**#**1..99999999999999999]]
With *-rg*, you can optionally specify the start or end of a trap ID range.

#1..99999999999999999
Selection of a trap ID is indicated by the leading # sign. openFT outputs those ADM traps that lie within the specified range.

-rg=#1..999999999999999999
The ADM trap with exactly this trap ID is output. If this ID does not exist (gaps in the numbering are possible), the following message is output:
`No ADM traps available for the selected criteria.`

-rg=#1..99999999999999999**-#**1..999999999999999999
The range starts with the ADM trap with the first specified trap ID and ends with the second specified trap ID.

If a number is specified with *-nb* that is smaller than the number of ADM traps in the range, the required number of records up to the end ID is output.

-rg=#1..999999999999999999**-**
The range starts with the ADM trap for the specified trap ID and ends with the most recent ADM trap.

If a number is specified with *-nb* that is smaller than the number of ADM traps in the period, the most recent ADM traps are output.

-rg=-#1..999999999999999999
The range ends with the ADM trap with the specified trap ID.

If a number is specified with *-nb* that is smaller than the number of ADM traps in the range, the required number of ADM traps up to the end ID is output.

-rg not specified
The trap ID range or the time period is not used as a selection criterion, in other words, output starts with the current (most recent) ADM trap.

-src=partner
-src allows you to specify that only those ADM traps are to be displayed that originate from a specific partner. You can specify the name from the partner list or specify the partner address.

-src not specified
The partner name is not used as a selection criterion.

-tt=[**fts**][**,**][**pts**][**,**][**ptu**][**,**][**rqc**][**,**][**rqf**][**,**][**rqs**]
-tt allows you to specify the type of ADM traps to be output. You can specify several values separated by commas:

fts All ADM traps are output that indicate that the asynchronous openFT has started (*FT-START) or stopped (*FT-STOP).

pts All ADM traps are output that indicate a status change of a partner system (*PART-STATE).

ptu All ADM traps are output that indicate that a partner system may not be reachable (*PART-UNREA).

rqs All ADM traps are output that indicate that the amount of requests in the request queue has reached a limit of at least 85% (*RQ-LIM-HIGH) or has fallen below a value of 80% (*RQ-LIM-LOW).

rqf All ADM traps are output that indicate failed transfer (*TRANS-FAIL).

rqc All ADM traps are output that indicate successful transfer (*TRANS-SUCC).

-tt not specified
The ADM trap type is not used as a selection criterion.

-nb=1.. 9999999 | **@a**
-nb allows you to specify the number of ADM traps to be output.

@a for *number*
-nb=@a outputs all ADM traps that meet the specified selection criteria.

-nb not specified
If *-nb* is not specified, the output will depend on whether *-rg* has also been specified or not:

– If *-rg* is specified, all ADM traps that meet the specified selection criteria are output (corresponds to *-nb=@a*).

– If *-rg* is not specified, then only one ADM trap is output (corresponds to *-nb=1*).

-l *-l* specifies that the ADM traps are to be output in detailed format.

-csv *-csv* specifies that the ADM traps are to be output in CSV format. The values in the output are separated by semicolons.

-csv must not be specified at the same time as *-l*.

Neither *-l* nor *-csv* specified
The ADM traps are output in the default short format.

6.33.1 Description of the output of ADM traps

When you output ADM traps using the *ftshwatp* command, you can select between a short, concise output format, a long, detailed output and finally, output in CSV format for further processing in external programs.

The ADM traps are identified by trap IDs. These IDs are assigned in ascending sequence. For technical reasons, the numbering sequence may contain gaps. The sequence of entries in the ADM trap log file does not always correspond to the temporal sequence in which the ADM traps occurred on the system concerned. Searching for records according to particular selection criteria can therefore take a long time, because it is in principle necessary to read in all the entries.

6.33.1.1 Short output format of an ADM trap

The last three ADM traps are output in this example:

```
$ftshwatp -nb=3
        TRAP-ID TYPE         DATE        TIME      SOURCE
             52 RQ-LIM-HIGH  2012-07-02  10:36:56  fileserv
             51 TRANS-FAIL   2012-07-02  10:36:48  FTSERV01
             50 PART-UNREA   2012-07-02  10:32:01  FTSERV01
```

Explanation

TRAP-ID
: Number of the ADM trap in the ADM trap log file, up to 18 digits.

TYPE Trap type.

Possible values:

FT-START
: Asynchronous openFT has started

FT-STOP
: Asynchronous openFT has stopped

PART-STATE
: Status change on a partner system

PART-UNREA
: Partner system possibly not reachable

RQ-LIM-HIGH
: Request queue has reached a filling level of at least 85%

RQ-LIM-LOW
: Request queue has fallen below a filling level of 80%

TRANS-SUCC
: Successful file transfer

TRANS-FAIL
: Failed file transfer

DATE Date on which the trap occurred.

TIME Time at which the trap occurred.

SOURCE
: Name of the partner on which the trap occurred.

6.33.1.2 Long output format of an ADM trap

Example for outputting the last two ADM traps in detailed format:

```
$ftshwatp -nb=2 -l
TRAP-ID      = 52 TYPE = RQ-LIM-HIGH  TIME = 2012-07-02 10:36:56
   SOURCE    = FTSERV01
   PARTNER   =                        PTN-STATE =
   TRANS-ID  =    RC   =              INITIATOR =
   FILENAME  =
   ERROR-MSG =
TRAP-ID      = 51 TYPE = TRANS-FAIL   TIME = 2012-07-02 10:36:48
   SOURCE    = admin001
   PARTNER   = PARTLINU               PTN-STATE =
   TRANS-ID  = 11 RC   = 2169         INITIATOR = user
   FILENAME  = order.txt
   ERROR-MSG = Request 11. Remote System: Transfer admission invalid
```

Explanation

TRAP-ID
: Number of the ADM trap in the ADM trap log file, up to 18 digits.

TYPE Trap type.

: The possible values are the same as for the short output format. See the description on page 285.

TIME Date and time at which the trap occurred.

SOURCE
: Name of the partner on which the trap occurred.

TRANS-ID
: Transfer ID of the transfer that triggered the trap.

RC Reason code of the transfer that triggered the trap.

INITIATOR
: User ID or location of the transfer that triggered the trap.

PARTNER
: Partner name of the transfer or partner that triggered the trap.

PTN-STATE
: Partner state of the partner that triggered the trap.

FILENAME
: Filename of the transfer that triggered the trap.

ERROR-MSG
: Message text of the transfer that triggered the trap.

6.34 ftshwc - Show openFT instances that can be remotely administered

ftshwc allows you to show the openFT instances that you are permitted to administer as remote administrator.

You can enter *ftshwc* both locally on the remote administration server and by remote administration using *ftadm* (see page 172):

– If you enter *ftshwc* locally on the remote administration server, the openFT instances are determined on the basis of the user ID under which you issue the *ftshwc* command.

– If you enter *ftshwc* via a remote administration request using *ftadm*, you must specify an FTAC transfer admission. The openFT instances are determined on the basis of the admission profile that belongs to this transfer admission.

ftshwc searches the configuration data on the remote administration server for openFT instances that are allowed to be remotely administered with the user ID or using this admission profile and outputs them.

If you are not permitted to remotely administer any instances, the following message is issued:

```
ftshwc: No instances available
```

Format

ftshwc -h |
[-rt=i | -rt=gi | -rt=ig]
[-csv]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-rt=i | **-rt=gi** | **-rt=ig**
-rt specifies what information is to be displayed.
You can specify the following: *i*, *gi* (default), *ig*

i Only information on instances is shown.

gi, ig Information on groups and instances is shown.

-csv *-csv* specifies that the data is to be output in CSV format.

-csv not specified
The data is output in default format.

6.34.1 Output format of ftshwc

Example of output in default format:

```
ftshwc

TYPE    = *GROUP        ACCESS =                  MODE =
   NAME = Hamburg
   DESC = Rechenzentrum Nord in Hamburg Wandsbek
TYPE    = *GROUP        ACCESS =                  MODE =
   NAME = Hamburg/HH1
   DESC = QA Rechenzentrum
TYPE    = *INSTANCE     ACCESS = FT+FTOP+FTAC     MODE = FTADM
   NAME = Hamburg/HH1/HHWSRV01
   DESC = Solaris 10
TYPE    = *INSTANCE     ACCESS = FT+FTOP+FTAC     MODE = FTADM
   NAME = Hamburg/HH1/HHWSRV02
   DESC = HP-11
TYPE    = *INSTANCE     ACCESS = FT+FTOP          MODE = LEGACY
   NAME = Hamburg/HH1/HHWSRV11
   DESC = Solaris 9
TYPE    = *GROUP        ACCESS =                  MODE =
   NAME = Hamburg/HH2
   DESC = Personalabteilung
TYPE    = *INSTANCE     ACCESS = FTOP             MODE = FTADM
   NAME = Hamburg/HH2/HHWSRV99
   DESC = Mainframe-System (BS2000/OSD)
```

Explanation

TYPE Specifies whether the item is a group or an openFT instance:

*GROUP
: Group

*INSTANCE
: openFT instance

ACCESS
: Only contains a value if *TYPE=*INSTANCE* and specifies what remote administration privileges the remote administrator has on this instance:

  FTOP Read FT access only (FT operator)

  FT Read and modify FT access. Corresponds to the permissions of an FT administrator.

  FTAC Read and modify FTAC access. Corresponds to the permissions of an FTAC administrator.

MODE

Only contains a value if *TYPE=*INSTANCE* and specifies the protocol that is used to administer this instance:

FTADM The instance is administered using the FTADM protocol.

LEGACY

The instance is administered using *ftexec*.

NAME

Pathname of the group or of the openFT instance.

In remote administration requests, you must always specify the name of the openFT instance as it is displayed here, i.e. as a complete pathname.

DESC

Description of the group or openFT instance.

6.35 ftshwd - Display diagnostic information

With the *ftshwd* command, you can display diagnostic information.

The diagnostic documents are used by the Maintenance and Diagnostic Service for error diagnosis.

Format

ftshwd

Description

The command has a number of options, but these are only significant for the Customer Service team.

Example

```
ftshwd

DATE     TIME    SSID  COMPONENT    LOCATION-ID       INFO
20120617 100921  FT    251/yfysequ  46/SwinsLwrite    ffffffff
20120617 100923  FTAC  39/yfslogg   1/WriteErr        ffffffff
```

Explanation of output:

DATE
: Date when the error occurred

TIME
: Time at which the error occurred

SSID
: Subsystem ID. Name of the subsystem that generated the diagnostic record.

COMPONENT
: Module number/name

LOCATION-ID
: Location in the code at which the error occurred.

INFO
: Error code

6.36 ftshwe - Display FT profiles and admission sets from a file

ftshwe stands for "show environment", i.e. display FT profiles and admission sets from a file. Using *ftshwe*, the FTAC administrator can display FT profiles and admission sets that were saved using the *ftexpe* command.

Format

ftshwe -h |
<file name 1..512>
[-u=<user ID 1..32>[,...,<user ID(100) 1..32>]]
[-pr=<profile name 1..8>[,...,<profile name(100) 1..8>] | -pr=@n]
[-as=y | -as=n]
[-l][-csv]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

file name
file name specifies the file from which the FT profiles and admission sets are to be displayed.

-u=user ID1[**,**user ID2][**,**user ID3]..
specifies the user IDs whose FT profiles and admission sets are to be displayed. You can specify up to 100 login names simultaneously.

If the specified user ID has no admission sets, only the standard admission set is displayed.

If you specify a non-existent login name for *user ID1*, the current standard admission set is displayed.

-u not specified
all FT profiles and admission sets are displayed.

-pr=profile name1[**,**profile name2][**,**profile name3]... | **-pr=@n**
specifies the FT profiles to be displayed (up to 100).

@n for *profile name*
no FT profiles are displayed.

-pr not specified
all FT profiles belonging to the user IDs specified in the -u parameter are displayed.

-as=y | **-as=n**
specifies whether or not admission sets are to be displayed.

y (default value)
all admission sets belonging to the login names specified in the *-u* parameter are displayed.

n no admission sets are displayed.

-l specifies that you wish to see the contents of the selected FT profiles.

-l not specified
displays only the names of the FT profiles. Markings also indicate whether or not an FT profile is privileged (*) and whether or not it is disabled (!).

-csv *-csv* specifies that the FT profiles and admission sets are to be output in CSV format. The values are output separated by semicolons. When *-csv* is specified, the output is always detailed (analogous to -l), regardless of whether or not -l is specified at the same time.

For details, see section “ftshwp” on page 405 and section “ftshwa” on page 387.

-csv not specified
The FT profiles and admission sets are output in the standard format.

6.37 ftshwk - Show properties of RSA keys

You can use the *ftshwk* command to output the properties of RSA keys. You can display the RSA keys of your own instance as well as the RSA keys of partners.

Format

ftshwk -h
[-own]
[-id=<identification 1..64> | -id=@a]
[-pn=<partner 1..200> | -pn=@a] |
[-exp=n | -exp=e | -exp=yyyymmdd | -exp=1..999]
[-csv]

Description

-h Outputs the command syntax on screen. Any specifications after *-h* are ignored.

-own Displays the key for your own instance.

-own must not be specified in combination with *-pn* or *-id*.

-id=identification | **-id=@a**
identification is the instance identification of the partner whose key is to be displayed. *-id* must not be specified in combination with *-pn* and *-own*.

@a Displays the installed keys of all partner systems.

-pn=partner | **-pn=@a**
partner is the name of the partner system in the partner list or the address of the partner system whose key is to be displayed.

-pn must not be specified in combination with *-id* and *-own*.

You will find detailed information on address specifications in the section “Specifying partner addresses” on page 68.

@a Displays the installed keys of all partner systems.

Neither *-id* nor *-pn* nor *-own* specified
Displays the keys of your own instance and the installed keys of all the partner systems.

-exp=n | **-exp=e** | **-exp=**yyyymmdd | **-exp=**1..999
Selects the keys on the basis of their expiration date.

n (no) Displays all partner keys that do not have an expiration date.

e (expired) Displays all partner keys that have already expired.

yyyymmdd
Displays all partner keys that expire at the latest at 00:00 local time on the specified date. For example, 20130101 displays all the keys that will become invalid by 00:00 on 01.01.2013.

1..999 Displays all partner keys that will expire within the specified number of days.

-exp not specified
The expiration date is not a selection criterion.

-csv *-csv* specifies that the key properties are to be output in CSV format. The values in the output are separated by semicolons.

-csv not specified
The key properties are output in the default format.

Example

You want to output the properties of all the keys:

```
ftshwk

CRE-DATE   EXP-DATE   KEY-LEN KEY-REF AUTHL IDENTIFICATION
2011-12-31            768     5       2
2011-12-31            1024    5       2
2011-12-31            2048    5       2
2012-01-31            1024    6       2
2012-02-29            2048    7       2
2011-03-28 2012-12-24 2048    7       2     MYOWNID.DOMAIN.NET
2011-07-12 EXPIRED    768     12      2     PC17QD.DOMAIN.NET
2011-05-14            1024    1036    1     PC27ABC.DOMAIN.NET
```

Explanation:

CRE-DATE
Date on which the key was generated.

EXP-DATE
Date on which the key expires, i.e. 00:00 on the specified day.
EXPIRED means that the key has already expired.

If there is no specification here then there is no expiration date.

KEY-LEN
: Key length in bit: 768, 1024 or 2048

KEY-REF
: Key reference

AUTHL Authentication level: 1 or 2

IDENTIFICATION
: Partner's instance ID. This field is left empty for keys belonging to your own instance.

6.38 ftshwl - Display log records and offline log files

With *ftshwl*, you can obtain information on all openFT requests logged up to now by openFT. In addition, you can output the names of the current log file and the offline log files.

If you are the FT, FTAC or ADM administrator, you can view log records of all user IDs. The log records are stored in the file *syslog.Lyymmdd.Lhhmmss*. This file is located in the *log* directory of the relevant openFT instance, see also “Instance directory” on page 26. *yymmdd* is the date (year, month, day) and *hhmmss* the time (hour, minute, second for GMT) at which the file was created. In the case of the default instance, the pathname is */var/openFT/std/log/syslog*.

For details on other instances, see the command *ftcrei* on page 184.

The log records are marked as FT, FTAC and ADM log records respectively, which means that you can determine the type of log record from the output.

For every request, there is an FTAC log record in which you can find the result of the FTAC admission check. For transfer requests, openFT logs whether it was actually able to execute this request in FT log records and for remote administration requests in ADM log records.

If no options are specified, openFT outputs the current log record. If options are specified, openFT outputs all log records up to the time specified in the command in reverse chronological order, i.e. starting from the most recent record to the oldest record.

The polling options allow you to specify that the output of new log records is to be repeated at regular intervals.

There are three types of output: short output, long output and CSV output (**C**haracter **S**eparated **V**alue).

Output is written to standard output.

Format

```
ftshwl  -h |
        [ <user ID 1..32> | @a]
        [ -lf=<file name1..512> | -tlf=yyyymmdd[hh[mm[ss]]] ]
        [ -plf=<0..3> ]
        [ -rg=[[[[yyyy]mm]dd]hhmm|#1..999999999999|0..999|:0..999][-
              [[[[yyyy]mm]dd]hhmm|#1..999999999999|0..999|:0..999]] ]
        [ -rt=[t][c][a] ]
        [ -ff=[t][m][r][d][a][C][D][M][I][f] ]
        [ -ini=l | -ini=r | -ini=lr | -ini=rl ]
        [ -pn=<partner 1..200> ]
        [ -fn=<file name 1..512> ]
        [ -rc=0..ffff | -rc=@f]
        [ -tid=1..2147483647 ]
        [ -gid=<globale request identification 1..4294967295> ]
        [ -adm=<administrator id 1..32> ]
        [ -ri=<routing info 1..200> ]
        [ -llf ]
        [ -nb=1..99999999 | -nb=@a ]
        [ -po=<polling interval 1..600>
              [ -pnr=<polling number 1..3600> ] ]
        [ -l ][ -csv ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

user ID | **@a**
: is used to specify the login name(s) for which log records are to be displayed. As the administrator, you can specify any login name.

  @a for *user ID*
  : FT, or FTAC or ADM administrators can display the log records for all login names.

user ID not specified
: Only the log records for the login name under which the command was entered are displayed.

-lf=file name | **-tlf=**yyyymmdd[hh[mm[ss]]]
: Selects the log file(s) whose log records or name are to be used. This means that you can also view offline log records.

  -lf=file name
  : The log file is selected based on its file name. You must specify the full relative or absolute path name. If no log file exists with the specified file name then an error message is output.

  -tlf=yyyymmdd[hh[mm[ss]]]
  : The log file is selected based on its creation time (local time!). The log file created at or before the specified time is selected. If more than one log file corresponds to the specified time then the next oldest log file is selected.

    You must at least specify the date as an 8-digit value indicating the year month and day. The year must be greater than or equal to 2000.
    You can specify the time (hhmmss) partially or not at all if you wish. "00" is added to replace any missing specifications. See also example 7.

Neither *-lf* nor *-tlf* specified
: The current log file is used.

-plf=number
: Specifies the number of preceding log files (0 to 3) that are to be selected in addition to the current file or the file specified with *-lf* or *-tlf*.

-plf not specified
: Selects only the current log file or the log file specified with *-lf* or *-tlf*.

If you omit the options *-plf* and *-lf* or *-tlf* then this corresponds to the behavior up to openFT V11.0.

-rg=[[[[yyyy]mm]dd]hhmm]**-**[[[[yyyy]mm]dd]hhmm]
: You can *-rg* to specify the start and/or end of a logging interval.

  [[[yyyy]mm]dd]hhmm
  : A 4-digit specification is interpreted as the time expressed in hours and minutes, a 6-digit specification as the day (date) and time in hours and minutes, an 8-digit specification as the month, day, and time in hours and minutes, and a 12-digit specification as the year, month, day, and time in hours and minutes. The largest possible value that can be specified as the date is 20380119 (January 19, 2038).
    openFT then displays all the log records written during the specified time period. The older time is taken to be the start time and the earlier time as the end time.
    If optional data (*[[[yyyy]mm]dd]*) is omitted, then it is automatically replaced by current values.

If you omit the limit after the dash, the current time is taken. If you omit the limit before the dash, the time of the first log record written is taken.

-rg=- Displays everything (same meaning as *-nb=@a*)

-rg=[[[yyyy]mm]dd]hhmm
If the minus sign is missing, the range is the exact minute specified. The largest possible value that can be specified as the date is 20380119 (January 19, 2038). If optional data (*[[[yyyy]mm]dd]*) is omitted, then it is automatically replaced by current values.

-rg=[**#**1..999999999999]**-**[**#**1..999999999999]
-rg is used to specify the start and/or end of a range of log IDs.

#1..999999999999
The selection of a log ID is indicated by the leading # character. openFT then displays all the log records which lie within the specified range.

If the log ID limit before the dash is omitted, the current ID is taken, and if the log ID limit after the dash is omitted, the ID of the first log record written is taken.

-rg=#1..999999999999
If the minus sign is omitted, the range is restricted to the specified log ID only.

-rg=[0..999][**-**[0..999]]
Here you specify with *-rg* a relative time period as a multiple of 24 hours (i.e. as a number of days). Note that the relative time period is calculated with an accuracy of one second from the current time. You have the following options (*d1* and *d2* 1 through 3 digits):

– *-rg=d1-d2* outputs all log records that are between *d1* and *d2* days old, irrespective of whether *d1* is larger or smaller than *d2*.

– *-rg=d1-* outputs all log records that are no more than *d1* days old.

– *-rg=-d2* outputs all log records that are at least *d2* days old.

-rg=[**:**0..999][**-**[**:**0..999]]
Here you specify with *-rg* a relative time period in minutes. You have the following options in this case (*m1* and *m2* 1 through 3 digits):

– *-rg=m1-:m2* outputs all log records that are between *m1* and *m2* minutes old, irrespective of whether *m1* is larger or smaller than *m2*.

– *-rg=:m1* (or **-***rg=:m1-*) outputs all log records that are no more than *m1* minutes old.

– *-rg=-:m2* outputs all log records that are at least *m2* minutes old.

-rg not specified
The range is not a selection criterion.

-rt=[**t**][**c**][**a**]

Defines which type of log record is to be displayed.
You may specify *t*, *c*, *a* and any combination of these values:

t The FT log records are displayed.

c The FTAC log records are displayed.

a The ADM log records are displayed.

-rt not specified

The record type is not a selection criterion.

-ff=[**t**][**m**][**r**][**d**][**a**][**C**][**D**][**M**][**l**][**f**]

Defines the FT function for which log records are to be output. Possible values are: *t*, *m*, *r*, *d*, *a*, *C*, *D*, *M*. *l*, *f* or any combination of these values.
The entries *m*, *r*, *d*, *a*, *C*, *D*, *M* and *l* are only reasonable for FTAC log records. The entry *f* is only reasonable for ADM log records. *t* is reasonable for all log records.

t All log records for the function "transfer files" are output.

m All log records for the function "modify file attributes" are output.

r All log records for the function "read directories" are output.

d All log records for the function "delete files" are output.

a All log records for the function "read file attributes" are output.

C All log records for the function "Create directory" are output.

D All log records for the function "Delete directory" are output.

M All log records for the function "Modify directory" are output.

l All log records for the function "inbound FTP access" are output. These log records are written if incorrect admission data (FTAC transfer admission or user ID/password) was specified for inbound FTP access.

f All ADM log records of the "Routing" function are output on the remote administration server. Output can be further restricted with the *-adm* and *-ri* options.

-ff not specified

The FT function is not a selection criterion.

-ini=l | **-ini=r** | **-ini=lr** | **-ini=rl**
Defines the initiator for which log records are to be output. Possible values are: *l*, *r*, *lr*, *rl*.

l (local) Only log records belonging to openFT requests issued locally are output.

r (remote) Only log records belonging to openFT requests issued remotely are output.

lr, rl The log records belonging to openFT requests issued locally and remotely are output.

-ini not specified
The initiator is not a selection criterion.

-pn=partner
Defines the partner system to which the log records are to be output. Partner is the name of the partner in the partner list or the address of the partner system. For details on address specifications, see section “Specifying partner addresses” on page 68

For the partner name, you can also use the wildcard symbols '*' (asterisk) and '?' (question mark). * stands for any string and ? stands for any single character.

-pn not specified
The partner system is not a selection criterion.

-fn=file name
Defines the file to which the log records are to be output. You can specify wildcards such as "*" (asterisk, i.e. any character string) and "?" (question mark, i.e. single character).

-fn not specified
The file name is not a selection criterion.

-rc=0..ffff | **@f**
Defines the reason code as a selection criterion for log record output.

0 .. ffff
All log records with a specified reason code are output.

@f All log records with reason codes other than 0000 are output. This criterion yields a list of log records for all requests terminated with error messages.

-rc not specified
The reason code is not a selection criterion.

-tid=request id
: *-tid* specifies the request number for which you want to output the log records.

-tid not specified
: The request id is not a selection criterion.

-gid=global request id
: With the -*gid*, you specify the global request ID for which you want to display log records. The global request ID is only relevant for inbound requests from openFT and FTAM partners. It is assigned by the initiator of the request (transfer ID) and is sent to the local system.

-gid= not specified
: The global request ID is not used as a selection criterion.

-adm=administrator id
: *-adm* specifies the administrator ID for which you want to output the ADM log records.

-adm not specified
: The administrator id is not a selection criterion.

-ri=routing info
: *-ri* specifies the routing information for which you want to output the ADM log records.

-ri not specified
: The routing info is not a selection criterion.

-llf outputs the names of log files. *-llf* is only permitted on its own or in combination with the options *-lf*, *-tlf*, *-plf*, *-csv* or *-h*. If any other combination is used then the command is rejected.

-llf without *-lf*, *-plf* or *-tlf* outputs the names of all the log files (current log file together with all the offline log files (up to a maximum of 1024)). To restrict the output, you can also specify *-lf*, *-plf* or *-tlf*, see also example 6.

-llf not specified
: Log records that correspond to the current selection criteria are displayed.

-nb=number | **@a**
: Defines the number of log records to be output.

@a for *number*
: All log records are output.

-nb not specified
: If *-rg* has also been specified, *-nb* is replaced by the value *-nb=@a*.

If *-rg* is also not specified, *-nb* is replaced by the value *-nb=1*.

-po=polling interval

The *polling interval* indicates the time between repetitions in seconds. On each repetition, all the new log records are filtered in accordance with the specified selection criteria and the detected records are output.

If you also specify *-pnr*, you can limit the number of times the data is output. If you specify *-po* without *-pnr*, output is repeated an unlimited number of times.

If repeated output has been started with the *-po* option (with or without *-pnr*), it can be canceled by an interrupt signal (e.g. Ctrl+C). In addition, the operation is canceled if an error occurs. When the asynchronous server is stopped, output is not interrupted but continues to be issued.

-po must not be specified in combination with *-lf*, *-llf*, *-plf*, *-tlf*, *-tid*, *-gid*, *-nb* or *-rg*.

Possible values: 1 through 600.

No log records should be deleted during polling as otherwise discontinuities in the output may appear!

-po not specified

The log records are output immediately and once only.

-pnr=polling number

-pnr specifies the number of repetitions.
-pnr can only be specified in conjunction with *-po*.

Possible values: 1 through 3600.

-pnr not specified

The output is repeated without restriction.

-l Defines that the log records are to be output in long form.

-l not specified

The log records are output in short form if *-csv* has not been specified.

-csv You can use *-csv* to specify that the log records are to be output in the CSV format. The values in the output are separated by semicolons.
If *-csv* is specified, output is always in long form (analogous to *-l*) regardless of whether or not *-l* has also been specified.

-csv not specified

The log records are output in the standard format, i.e. in abbreviated form if *-l* is not specified and in detailed form if *-l* is specified.

Examples

The following examples each output the log records for the user's own ID. If you are an FT, FTAC or ADM administrator and want to output the log records for all user IDs, you must also specify *@a*.

1. All log records that are more than two days (48 hours) old are output:

   ```
   ftshwl -rg=-2
   ```

2. All log records that are more than 15 minutes old but less than 30 minutes old are output:

   ```
   ftshwl  rg=:15-:30
   ```

3. All log records that are less than 30 minutes old are output:

   ```
   ftshwl -rg=:30
   ```

4. All log records that are more than 30 minutes old are output:

   ```
   ftshwl -rg=-:30
   ```

5. The last 10 log records where FTAC checks failed (reason code not equal to 0) are output:

   ```
   ftshwl -rc=@f -rt=c -nb=10
   ```

6. The name of the current log file and the names of the two preceding offline log files are to be output:
   ```
   ftshwl -llf -plf=2
   ```

7. Output of 100 log records from the log file that was created on or before 24.02.2012 00:00:
   ```
   ftshwl -tlf=20120224 -nb=100
   ```

   Note

   -tlf=20120224 is extended to *-tlf=20120224000000*. If, for example, there are three log files with the creation dates 20120224 13:30:00, 20120217 10:00:00 and 20120210 08:00:00, then the file with the date 20120217 10:00:00 is taken as the next oldest file.

6.38.1 Description of log record output

Log records can be displayed using the openFT Explorer or by using the *ftshwl* command. You can choose between a short overview, detailed information or, if further processing is to be performed with external programs, output in the CSV format.

The log records are identified by log IDs. The log IDs are assigned in ascending order, but for technical reasons the numbering is not contiguous (i.e. there may be gaps).

6.38.1.1 Logging requests with preprocessing/postprocessing

For security reasons, only the first 32 characters (or 42 characters in the case of *ftexecsv* preprocessing) of a preprocessing or postprocessing command are transferred to the log record. By arranging the call parameters appropriately or by inserting blanks, you can influence which command parameters do not appear in the log.

6.38.1.2 Short output format of a FT or FTAC log records

Example: The option *-rt=tc* causes only FT and FTAC log records to be output.

```
$ftshwl -rt=tc -nb=12
TYP LOG-ID TIME     RC    PARTNER  INITIAT. PROFILE  USER-ADM FILENAME
2012-05-05
CA    8273 09:16:07 0000 >PARTLINU *REMOTE  pr1      user1    file.10
CA    8272 09:16:07 0000 >PARTLINU user1             user1
CD    8271 09:15:30 0000 <PARTLINU *REMOTE  pr1      user1    file.new
CD    8270 09:15:30 0000 <PARTLINU user1             user1
CM    8269 09:15:03 0000 <PARTLINU *REMOTE  pr1      user1    file.rem
CM    8268 09:15:03 0000 <PARTLINU user1             user1    file.new
CR    8267 09:14:14 0000 >PARTLINU *REMOTE  pr1      user1    .
CR    8266 09:14:14 0000 >PARTLINU user1             user1
T     8265 09:13:50 0000 >PARTLINU user1             user1    file.10
T     8264 09:13:50 0000 <PARTLINU *REMOTE           user1    file.rem
C     8263 09:13:49 0000 <PARTLINU *REMOTE  pr1      user1    file.rem
C     8262 09:13:49 0000 >PARTLINU user1             user1    file.10
```

Explanation

TYP Comprises three columns. The first column specifies whether the log record is an FT or FTAC log record:

- T FT log record
- C FTAC log record

The second and third column identify the FT function:

- _ (empty): transfer file
- A read file attributes (only in the FTAC log record)
- D delete file (only in the FTAC log record)
- C create file (only in the FTAC log record)
  possible only for transfer requests issued in the remote partner system
- M modify file attributes (only in the FTAC log record)
- R read directory (only in the FTAC log record)
- CD create directory (only in FTAC log record)
- DD delete directory (only in FTAC log record)
- MD modify directory attributes (only in FTAC log record)
- L Login: Failed inbound FTP access (only in FTAC log record)

LOG-ID
Log record number

TIME
specifies time when the log record was written

RC Reason code. Specifies whether a request was successful (RC=0) or if not, why it was rejected or cancelled. Additional information on the reason code is available using the *fthelp* command.

PARTNER
Provides information about the partner system involved. The name in the partner list or the address of the partner system, possibly truncated to 8 characters, or the name under which the partner system is entered in the TNS is output.

The name or address of the partner system is preceded by an identifier to indicate the direction of the request.

> The request is sent to partner system. This transfer direction is specified for a
>
> – send request
> – a request to display file attributes
> – a request to display directories

< The request is sent to local system. This transfer connection is specified for

– a receive request
– a request to modify file attributes
(When a FTAM partner modifies the access rights of a local file, two log records are written. No direction is specified in front of PARTNER in this case.)
– a request to delete files

INITIAT.
Request initiator. If initiated in the remote system: *REMOTE.

PROFILE
Name of the profile used for file transfer (only in FTAC log record).

USER-ADM
Login name to which the requests in the local system refer.

If a login name longer than 8 bytes was specified, the first seven bytes are output, followed by an asterisk (*).

FILENAME
Local file name

6.38.1.3 Short output format of an ADM log record

In the following examples, the option *-rt=a* causes only ADM log records to be output.

1. Output ADM log records on a client:

```
ftshwl ftadmin -rt=a -nb=5
TYP LOG-ID TIME     RC    PARTNER  INITIAT. PROFILE  USER-ADM FILENAME
2012-05-19
A       39 04:30:35 0000 <flexthom ftadmin           ftadmin
A       36 04:30:15 0000 <flexthom ftadmin           ftadmin
A       33 04:29:49 0000 <flexthom ftadmin           ftadmin
A       30 04:28:15 0000 <flexthom ftadmin           ftadmin
A       27 04:22:56 0000 <flexthom ftadmin           ftadmin
```

2. Output ADM log record on the administered openFT instance:

```
ftshwl -rt=a
TYP LOG-ID TIME     RC    PARTNER  INITIAT. PROFILE  USER-ADM FILENAME
2012-05-19
A     2575 13:30:15 0000 >ftadm:/* *REMOTE  adminrem admin001
```

3. Output routing ADM log record on the remote administration server:

```
ftshwl -rt=a -ff=f
TYP LOG-ID TIME     RC    PARTNER  INITIAT. PROFILE  USER-ADM FILENAME
2012-05-19
AF     396 13:22:54 0000 >Testrech *REMOTE  adminacc admin002
```

Explanation

The following differences apply to ADM log records compared with FT or FTAC log records:

– The value *A* is output for an ADM log record in the TYP column. In the case of ADM log records with routing information on the remote administration server (*ftshwl -ff=f*), the value *F* is also shown in column 2.

– The FILENAME column is empty for ADM log records.

6.38.1.4 Long output format of an FT log record

The log records with the numbers 103 and 404 are to be output in long form:

```
ftshwl @a -rg=#103 -l
LOGGING-ID  = 103       RC      = 2155        TIME     = 2012-05-23 10:53:22
   TRANS    = FROM      REC-TYPE= FT          FUNCTION = TRANSFER-FILE
   PROFILE  =           PCMD    = NONE        STARTTIME= 2012-05-23 10:53:20
   TRANS-ID = 65539     WRITE   = REPLACE     REQUESTED= 2012-05-23 10:53:20
   TRANSFER =         0 kB                    CCS-NAME = ISO88591
                                              CHG-DATE = SAME
   SEC-OPTS = ENCR+DICHK, RSA-2048 / AES-256
   INITIATOR= smith
   USER-ADM = smith
   PARTNER  = FTSERV01
   FILENAME = test01
   ERRINFO  = CreateFile(Attr.): The system cannot find the file specified

ftshwl @a -rg=#404 -l

LOGGING-ID  = 404       RC      = 0000        TIME     = 2012-07-06 13:37:17
   TRANS    = FROM      REC-TYPE= FT          FUNCTION = TRANSFER-FILE
   PROFILE  =           PCMD    = NONE        STARTTIME= 2012-07-06 13:37:16
   TRANS-ID = 262164    WRITE   = REPLACE     STORETIME= 2012-07-06 13:37:17
   TRANSFER =         5 kB                    CCS-NAME =
   SEC-OPTS = ENCR+DICHK+RAUTH, RSA-2048 / AES-128
   INITIATOR= *REMOTE                         GLOB-ID  = 67017
   USER-ADM = smith
   PARTNER  = mc122.othernet.local
   FILENAME = example
```

Explanation

LOGGING-ID
: Log record number; up to twelve characters in length

TRANS
: Transfer direction

  TO Transfer direction to the partner system. This transfer direction is specified for

  – a send request
  – a request to display the file attributes
  – a request to display the directories

FROM
Transfer direction to the local system. This transfer direction is specified for

- a receive request
- a request to modify the file attributes
- a request to delete files

PROFILE
Name of profile used

TRANS-ID
Request number

TRANSFER
Number of bytes transferred

SEC-OPTS
Security options used during transfer

ENCR Encryption of the request description

DICHK Data integrity check of the request description

DENCR Encryption of the transferred file content

DDICHK Data integrity check of the transferred file content

LAUTH Authentication of the local system in the remote system (authentication level 1)

LAUTH2 Authentication level of the local system in the remote system (authentication level 2)

RAUTH Authentication of the remote system in the local system (authentication level 1)

RAUTH2 Authentication level of the remote system in the local system (authentication level 2)

RSA-nnn
Length of the RSA key used for the encryption

AES-128 / AES-256 / DES
The encryption algorithm used

INITIATOR
Request initiator. If initiated in the local system: login name. If initiated in the remote system: *REMOTE

USER-ADM
Login name to which the requests in the local system refer

PARTNER
: Identifies the partner system in question.
The name in the partner list or the address of the partner system, possibly truncated to 8 characters, or the name under which the partner system is entered in the TNS is output.
In the case of requests issued from a remote computer, it is also possible for *%strange* to be output followed by a part of the address of the partner system if the partner system is not entered in the TNS and TCP/IP-RFC1006 was not used as the transport system.

FILENAME
: Local file name

ERRINFO
: Additional information on the error message if an error occurred during a transfer.

RC
: Reason code. Specifies whether a request was successful (RC=0) or if not, why it was rejected or cancelled. You can obtain further information with the *fthelp* command.

REC-TYPE
: Specifies whether the log record is an FT log record.

PCMD
: Indicates whether follow-up processing was specified and started. Possible values:

  NONE
  : No follow-up processing specified

  STARTED
  : Follow-up processing was started (contains no information about the successful completion of follow-up processing!).

  NOT-STARTED
  : Follow-up processing could not be started.

WRITE
: Write mode. The field is assigned a value only for outbound requests; for inbound requests, it contains a blank. Possible values:

  NEW
  : A new file is created. If a file with this name already exists, file transfer is aborted.

  EXT
  : An existing file is extended, otherwise a new is created.

  REPLACE
  : An existing file is overwritten. If it does not already exist, it is created.

TIME
: Specifies time when log record was written

FUNCTION
: FT function

  TRANSFER-FILE
  : Transfer file

STARTTIME
: Indicates the start time of the request.

STORETIME
: If the request was submitted in the remote system then the time of the entry in the request queue is displayed here.

REQUESTED
: When initiative in the local system, the time of issue of the request is shown here.

Depending on the initiator of the request (local or remote), either STORETIME or REQUESTED is output but never both together.

CCS-NAME
: Name of the character set used to code the local file.

CHG-DATE
: Specifies whether the change date of the send file is taken over for the receive file.

  SAME The modification date of the send file is taken over.

GLOB-ID
: Global request identification, displayed in the case of inbound requests from openFT and FTAM partners (INITIATOR=REMOTE). This corresponds to the request identification (=TRANSFER-ID) on the initiator system.

6.38.1.5 Long output format of an FTAC log record

The log record with log record number 5172 is to be output in long form:

```
ftshwl @a -rg=#5172 -l
LOGGING-ID  = 00005172  RC      = 0000        TIME     = 2012-04-03 09:38:06
   TRANS    = TO        REC-TYPE= FTAC        FUNCTION = TRANSFER-FILE
   PROFILE  = remadmin  PRIV    = NO
   INITIATOR= *REMOTE
   USER-ADM = thomasw
   PARTNER  = angel.domain1.de
   FILENAME = |ftexecsv ftshwo -tn -a -u -ccs=ISO8859-1
```

Explanation

LOGGING-ID
: Log record number, up to twelve characters in length

TRANS
: Transfer direction

  TO Transfer direction to partner system. This transfer direction is specified for

  – a send request
  – a request to display the file attributes
  – a request to display the directories

  FROM
  : Transfer direction to local system. This transfer direction is specified for

    – a receive request
    – a request to modify the file attributes
    – a request to delete files

  BOTH
  : The request direction is to the partner system and to the local system. When an FTAM partner modifies the access rights of a local file, two log records are written. The direction BOTH is specified in each.

PROFILE
: Name of the profile used

INITIATOR
: Request initiator. If initiated in the local system: login name. If initiated in the remote system: *REMOTE

USER-ADM
: Login name to which the requests in the local system refer

PARTNER
: Identifies the partner system in question.
The name in the partner list or the address of the partner system, possibly truncated to 8 characters, or the name under which the partner system is entered in the TNS is output.
In the case of requests issued from a remote computer, it is also possible for *%strange* to be output followed by a part of the address of the partner system if the partner system is not entered in the TNS and TCP/IP-RFC1006 was not used as the transport system.

FILENAME
: Local file name

RC
: Reason code. Specifies whether a request was successful (RC=0) or if not, why it was rejected or cancelled. You can use the *fthelp* command to obtain further information.

REC-TYPE
: Specifies whether the log record is an FTAC log record.

PRIV
: Specifies whether or not the FT profile being used is privileged

TIME
: Specifies time when the log record was written

FUNCTION
: FT function

  TRANSFER-FILE
  : Transfer file

  READ-FILE-ATTR
  : Read file attributes

  DELETE-FILE
  : Delete file

  CREATE-FILE
  : Create file (possible only in requests submitted in the remote partner system)

  MODIFY-FILE-ATTR
  : Modify file attributes

  READ-FILE-DIR
  : Read directories

  CREATE-FILE-DIR
  : Create file directory

DELETE-FILE-DIR
: Delete file directory

MODIFY-FILE-DIR
: Modify file directory

LOGIN
: Login: Inbound FTP access.
This log record is written if incorrect admission data was specified for inbound FTP access.

6.38.1.6 Long output format of an ADM log record

In the following examples, the option *-rt=a* causes only ADM log records to be output.

1. ADM log record on a client:

```
ftshwl -rt=a -l
LOGGING-ID  = 27        RC      = 0000        TIME     = 2012-05-19 04:22:56
   TRANS    = FROM      REC-TYPE= ADM         FUNCTION = REM-ADMIN
   TRANS-ID = 190845    PROFILE =
   SEC-OPTS = ENCR+DICHK, RSA-768 / AES-256
   INITIATOR= ftadmin
   USER-ADM = ftadmin
   PARTNER  = flexthom
   ADM-CMD  = ftshwo
   ADMIN-ID =
   ROUTING  = Muenchen/Jonny
```

2. ADM log records on the remote administration server:

```
ftshwl -rt=a -l -nb=3
LOGGING-ID  = 400       RC      = 0000        TIME     = 2012-05-19 13:22:56
   TRANS    = TO        REC-TYPE= ADM         FUNCTION = REM-ADMIN
   TRANS-ID = 65608     PROFILE = adminacc
   SEC-OPTS = ENCR+DICHK, RSA-2048 / AES-256
   INITIATOR= *REMOTE
   USER-ADM = admin002
   PARTNER  = ftadm://cog2-test-eng.homenet.de
   ADM-CMD  = ftshwo
   ADMIN-ID = Hugo
   ROUTING  = Munich/Jonny

LOGGING-ID  = 399       RC      = 0000        TIME     = 2012-05-19 13:22:55
   TRANS    = FROM      REC-TYPE= ADM         FUNCTION = REM-ADMIN
   TRANS-ID = 152973    PROFILE =
   SEC-OPTS = ENCR+DICHK, RSA-2048 / AES-256
   INITIATOR= admin002
   USER-ADM = admin002
   PARTNER  = Test0001
   ADM-CMD  = ftshwo
   ADMIN-ID =
   ROUTING  =
LOGGING-ID  = 396       RC      = 0000        TIME     = 2012-05-19 13:22:54
   TRANS    = TO        REC-TYPE= ADM         FUNCTION = REM-ADMIN-ROUT
   TRANS-ID =           PROFILE = adminacc
   SEC-OPTS =
   INITIATOR= *REMOTE
   USER-ADM = admin002
   PARTNER  = Test0001
   ADM-CMD  = ftshwo
   ADMIN-ID = Hugo
   ROUTING  = Munich/Jonny
```

3. ADM log record on the administered openFT instance:

```
ftshwl -rt=a -l
LOGGING-ID  = 2571      RC      = 0000        TIME     = 2012-05-19 13:29:49
   TRANS    = TO        REC-TYPE= ADM         FUNCTION = REM-ADMIN
   TRANS-ID = 334030    PROFILE = adminrem
   SEC-OPTS = ENCR+DICHK, RSA-2048 / AES-256
   INITIATOR= *REMOTE
   USER-ADM = admin001
   PARTNER  = ftadm://flexthom.homenet.de
   ADM-CMD  = ftshwl
   ADMIN-ID =
   ROUTING  =
```

Explanation

LOGGING-ID
: Log record number, up to twelve characters in length

RC
: Reason code of the request.

TIME
: Specifies time when the log record was written

REC-TYPE
: ADM is always output here for ADM log records

FUNCTION
: Administration function executed:

REM-ADMIN
: Execute remote administration request

REM-ADMIN-ROUT
: Check admission for remote administration request and forward remote administration request to the openFT instance to be administered if the admission check is successful

TRANS-ID
: Number of the administration request

PROFILE
: Name of the profile used

SEC-OPTS
: Security options used during transfer:

ENCR
: Encryption of the request description

DICHK
: Data integrity check of the request description

DENCR
: Encryption of the transferred file content

DDICHK
: Data integrity check of the transferred file content

  LAUTH
  : Authentication of the local system in the remote system (authentication level 1)

  LAUTH2
  : Authentication level of the local system in the remote system (authentication level 2)

  RAUTH
  : Authentication of the remote system in the local system (authentication level 1)

  RAUTH2
  : Authentication level of the remote system in the local system (authentication level 2)

  RSA-nnn
  : Length of the RSA key used for the encryption

  AES-128 / AES-256 / DES
  : The encryption algorithm used

INITIATOR
: Request initiator. If initiated in the local system: login name. If initiated in the remote system: *REMOTE

USER-ADM
: User ID to which the remote administration request refers in the local system

PARTNER
: Partner system involved. Depending on the location to which the ADM log record was written, the following is output:

  – Client: Name/address of the remote administration server
  – Remote administration server (inbound): Name/address of the client
  – Remote administration server (outbound): Name/address of the openFT instance to be administered
  – Administered openFT instance: Name/address of the remote administration server

ADM-CMD
: Administration command without parameters

ADMIN-ID
: Administrator ID under which the request is processed on the remote administration server. In the case of ADM log records on a client, this field is empty.

ROUTING
: Routing information on the openFT instance to be administered

6.38.2 Reason codes of the logging function

The FTAC log records contain a reason code which indicates whether an request was accepted after the admission check successfully and if not, why it was rejected.

In ADM log records, the reason code specifies why a remote administration request was not executed.

You can use the *fthelp* command to output the message text associated with the code number (see page 212):

`fthelp` *code-number*

In many codes, the last three digits correspond to the number of the associated openFT message.

In addition, there are a certain number of codes which do not correspond to openFT messages (see openFT User Guide). These are listed in the tables below:

RC	Reason
0000	Request successfully completed.
1001	Request rejected. Invalid transfer admission
1003	Request rejected. Transfer direction not permissible
1004	Request rejected. Illegal partner
1006	Request rejected. Violation of file name restriction
100f	Request rejected. Violation of success processing restriction
1010	Request rejected. Violation of failure processing restriction
1011	Request rejected. Violation of write mode restriction
1012	Request rejected. Violation of FT function restriction
1014	Request rejected. Violation of data encryption restriction
2001	Request rejected. Syntax error on file name extension
2004	Request rejected. Overall length of follow-up processing exceeds 1000 characters
3001	Request rejected. Invalid user identification
3003	Request rejected. Invalid password
3004	Request rejected. Transfer admission locked
3011	Request rejected. Violation of user outbound send level
3012	Request rejected. Violation of user outbound receive level
3013	Request rejected. Violation of user inbound send level
3014	Request rejected. Violation of user inbound receive level

RC	Reason
3015	Request rejected. Violation of user inbound processing level
3016	Request rejected. Violation of user inbound file management level
3021	Request rejected. Violation of ADM outbound send level
3022	Request rejected. Violation of ADM outbound receive level
3023	Request rejected. Violation of ADM inbound send level
3024	Request rejected. Violation of ADM inbound receive level
3025	Request rejected. Violation of ADM inbound processing level
3026	Request rejected. Violation of ADM inbound file management level

RC	Reason
7001	The administrator ID is invalid. It was not possible to determine a valid administrator ID from the user ID or the profile name in the configuration data of the remote administration server.
7002	The routing information is invalid. The specified openFT instance specified in the routing information could not be found in the configuration data of the remote administration server.
7003	The specified remote administration command is invalid. The remote administration server rejects the specified command because it is not a supported remote administration command.
7101	Infringement against the access rights list. On checking the access rights, the system identified that the administrator ID has not been assigned the necessary rights in the configuration data of the remote administration server to be able to execute the valid remote administration command on the specified openFT instance.
7201	Infringement against the maximum command length. In particular in the case of BS2000 commands, the remote administration server replaces the shortest command names, which are guaranteed by openFT, by the full command names. If this replacement of the command name causes the entire remote administration command to become longer than the maximum command length of 8192 characters, the command is rejected.

6.39 ftshwm - Display monitoring values of openFT operation

The *ftshwm* command allows you to output the current monitoring values from openFT operation. In order to do this, the FT administrator must have activated monitoring (*ftmodo -mon=n* command) and the asynchronous openFT server must be running.

Format

ftshwm -h |
[-ty]
[-raw]
[-po=<polling interval 1..600> [-pnr=<polling number 1..3600>]]
[-csv]
[<name 1..12> [... <name(100) 1..12>]| @a]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-ty The types and scaling factors are to be output in place of the monitoring values and metadata.

The metadata type can be *TIME (timestamp) or *STRING (text output of the chosen selection).

A monitoring value can have one of the following types:
INT, BOOL or PERCENT (integer, on/off value or percentage). In the case of integer values, the scaling factor may be specified in brackets: INT(<scaling factor>).

The scaling factor of a monitoring value is only significant for output in CSV format. In this case, it is the number by which the value shown must be divided in order to obtain the real value.

-raw must not be specified at the same time.

-raw Monitoring values are to be output as unedited raw data. This option is intended to be used in conjunction with external programs for further processing. The option must not be specified in conjunction with *-ty*. Monitoring values of the object *Duration* are not output.

If the specification is not used, the data is output in print-edited form.

The following section “Description of the monitoring values” on page 324 contains a table with notes that show what values are output when the *-raw* option is specified or is not specified and how the values are to be interpreted depending on this option.

-po=polling interval
: Data is to be output initially after the specified polling interval in seconds has elapsed and then repeated at this interval.

  If you also specify *-pnr*, you can limit the number of times the data is output. If you specify *-po* without *-pnr*, output is repeated an unlimited number of times.

  If repeated output has been started with the *-po* option (with or without *-pnr*), it can be cancelled by an interrupt signal. Output is also cancelled in the event of an error, when the asynchronous openFT is terminated, or when monitoring is terminated.

  Possible values: 1 through 600.

-po not specified
: The monitoring values are output immediately and once only.

-pnr=polling number
: *-pnr* specifies the number of times data is output. *-pnr* can only be specified in conjunction with *-po*.

  Possible values: 1 through 3600.

-csv The information is to be output in CSV format. First, the short names of the monitoring values are output in one row as the field names. This is followed by a row containing the monitoring values or their types and scaling factors as decimal numbers.

  You can limit the scope of the output by specifying individual monitoring values that are significant for you.

name [name ...] | **@a**
: The specified monitoring value or, if *-ty* is specified, the type and scaling factor associated with the named value is to be output.

  name must be one of the short names of the monitoring values as they appear in the CSV header. You can specify up to 100 names separated by blanks.

  @a for *name*
  : All openFT monitoring values or the types and scaling factors of all openFT monitoring values are to be output.

name not specified
: A predefined default set of monitoring values is output (see the section “Description of the monitoring values” on page 324).

6.39.1 Description of the monitoring values

The table below shows all the monitoring values output with the option *@a*. You can instead specify a list of any of the monitoring values shown in the table.

You can use the openFT Monitor to display the monitoring values for openFT operation. You call the openFT Monitor by means of the *ftmonitor* command see openFT User Manual

The first two letters of the name indicate the data object that the monitoring value belongs to:

– Th = Throughput
– Du = Duration
– St = State

The second component of the name indicates the performance indicator, e.g. *Netb* for net bytes. In the case of monitoring values for the *Throughput* or *Duration* data object, the last 3 letters of the name indicate the types of requests from which the monitoring value originates, e.g.

– Ttl = FT Total
– Snd = FT Send requests
– Rcv = FT Receive requests
– Txt = Transfer of text files
– Bin = Transfer of binary files
– Out = FT Outbound
– Inb = FT Inbound

If monitoring is deactivated for all partners (*ftmodo -monp=*), only the following values are populated:

Status: StCLim, StCAct, StRqLim, StRqAct, StOftr, StFtmr, StFtpr, StTrcr

All the other values are set to 0.

Name	Meaning	Output prepared (formatted)	Output not prepared (raw)
ThNetbTtl	Throughput in net bytes: Number of bytes transferred	Number of bytes per second	Bytes, accumulated
ThNetbSnd	Throughput in net bytes (send requests): Number of bytes transferred with send requests7	Number of bytes per second	Bytes, accumulated
ThNetbRcv	Throughput in net bytes (receive requests): Number of bytes transferred with receive requests	Number of bytes per second	Bytes, accumulated
ThNetbTxt [1)]	Throughput in net bytes (text files): Number of bytes transferred when transferring text files	Number of bytes per second	Bytes, accumulated
ThNetbBin [1)]	Throughput in net bytes (binary files): Number of bytes transferred when transferring binary files	Number of bytes per second	Bytes, accumulated
ThDiskTtl	Throughput in disk bytes: Number of bytes read from files or written to files with transfer requests	Number of bytes per second	Bytes, accumulated
ThDiskSnd	Throughput in disk bytes (send requests): Number of bytes read from files with send requests	Number of bytes per second	Bytes, accumulated
ThDiskRcv	Throughput in disk bytes (receive requests): Number of bytes written to files with receive requests	Number of bytes per second	Bytes, accumulated
ThDiskTxt [1)]	Throughput in disk bytes (text files): Number of bytes read from text files or written to text files with transfer requests	Number of bytes per second	Bytes, accumulated
ThDiskBin [1)]	Throughput in disk bytes (binary files): Number of bytes read from binary files or written to binary files with transfer requests	Number of bytes per second	Bytes, accumulated
ThRqto	openFT requests: Number of openFT requests received	Number per second	Number, accumulated
ThRqft [1)]	File transfer requests: Number of file transfer requests received	Number per second	Number, accumulated
ThRqfm [1)]	File management requests: Number of file management requests received	Number per second	Number, accumulated
ThSuct	Successful requests: Number of successfully completed openFT requests	Number per second	Number, accumulated

Name	Meaning	Output prepared (formatted)	Output not prepared (raw)
ThAbrt	Aborted requests: Number of aborted openFT requests	Number per second	Number, accumulated
ThIntr	Interrupted requests: Number of interrupted openFT requests	Number per second	Number, accumulated
ThUsrf	Requests from non-authorized users: Number of openFT requests in which the user check was terminated with errors	Number per second	Number, accumulated
ThFoll [1)]	Follow-up processing operations started: Number of follow-up processing operations started	Number per second	Number, accumulated
ThCosu [1)]	Connections established: Number of connections successfully established	Number per second	Number, accumulated
ThCofl	Failed connection attempts: Number of attempts to establish a connection that failed with errors	Number per second	Number, accumulated
ThCobr	Disconnections: Number of disconnections as a result of connection errors	Number per second	Number, accumulated
DuRqtlOut [1)]	Maximum request duration Outbound: Maximum request duration of an outbound request	Milliseconds [2)]	-
DuRqtlInb [1)]	Maximum request duration Inbound: Maximum request duration of an inbound request	Milliseconds [2)]	-
DuRqftOut [1)]	Maximum request duration Outbound transfer: Maximum duration of an outbound file transfer request	Milliseconds [2)]	-
DuRqftInb [1)]	Maximum request duration Inbound transfer: Maximum duration of an inbound file transfer request	Milliseconds [2)]	-
DuRqfmOut [1)]	Maximum request duration Outbound file management: Maximum duration of an outbound file management request	Milliseconds [2)]	-

Name	Meaning	Output prepared (formatted)	Output not prepared (raw)
DuRqfmInb [1)]	Maximum request duration Inbound file management: Maximum duration of an inbound file management request	Milliseconds [2)]	-
DuRqesOut [1)]	Maximum outbound request waiting time: Maximum waiting time before an outbound request is processed (for requests without a specific start time)	Milliseconds [2)]	-
DuDnscOut [1)]	Maximum duration of an outbound DNS request: Maximum time an outbound openFT request was waiting for partner checking	Milliseconds [2))]	-
DuDnscInb [1)]	Maximum duration of an inbound DNS request: Maximum time an inbound openFT request was waiting for partner checking	Milliseconds [2)]	-
DuConnOut [1)]	Maximum duration of establishment of a connection: Maximum time between requesting a connection and receiving confirmation of a connection for an outbound openFT request	Milliseconds [2)]	-
DuOpenOut [1)]	Maximum file open time (outbound): Maximum time an outbound openFT request required to open the local file	Milliseconds [2)]	-
DuOpenInb [1)]	Maximum file open time (inbound): Maximum time an inbound openFT request required to open the local file	Milliseconds [2)]	-
DuClosOut [1)]	Maximum file close time (outbound): Maximum time an outbound openFT request required to close the local file	Milliseconds [2)]	-
DuClosInb [1)]	Maximum file close time (inbound): Maximum time an inbound openFT request required to close the local file	Milliseconds [2)]	-
DuUsrcOut [1)]	Maximum user check time (outbound): Maximum time an outbound openFT request required to check the user ID and transfer admission	Milliseconds [2)]	-
DuUsrcInb [1)]	Maximum user check time (inbound): Maximum time an inbound openFT request required to check the user ID and transfer admission	Milliseconds [2)]	-

Name	Meaning	Output prepared (formatted)	Output not prepared (raw)
StRqas	Number of synchronous requests in the ACTIVE state	Average value [3]	Current number
StRqaa	Number of asynchronous requests in the ACTIVE state	Average value [3]	Current number
StRqwt	Number of requests in the WAIT state	Average value [3]	Current number
StRqhd	Number of requests in the HOLD state	Average value [3]	Current number
StRqsp	Number of requests in the SUSPEND state	Average value [3]	Current number
StRqlk	Number of requests in the LOCKED state	Average value [3]	Current number
StRqfi [1]	Number of requests in the FINISHED state	Average value [3]	Current number
StCLim	Maximum number of connections: Upper limit for the number of connections established for asynchronous requests.	Value currently set	
StCAct	Number of occupied connections for asynchronous requests	Share of StCLim in % [4]	Current number
StRqLim	Maximum number of requests: Maximum number of asynchronous requests in request management	Value currently set	
StRqAct	Entries occupied in request management	Share of StRqLim in % [4]	Current number
StOftr	openFT Protocol activated/deactivated	ON (activated) OFF (deactivated)	
StFtmr	FTAM protocol activated/deactivated	ON (activated) OFF (deactivated)	
StFtpr	FTP protocol activated/deactivated	ON (activated) OFF (deactivated)	
StTrcr [1]	Trace activated/deactivated	ON (activated) OFF (deactivated)	

[1] Output only if @a is specified

[2] Maximum value of the monitoring interval (= time elapsed since the last time the monitoring values were queried or since the start of monitoring).

[3] Average value of the monitoring interval (= time elapsed since the last time the monitoring values were queried or since the start of monitoring). Format: n.mm, where n is an integer and mm are to be interpreted as decimal places.

[4] If the reference value is reduced in live operation, it is possible for the value output to lie above 100 (%) temporarily.

Example

```
ftshwm
```

```
openFT(std)   Monitoring (formatted)
MonOn=2012-02-17 15:36:12 PartnerSel=OPENFT  RequestSel=ONLY-ASYNC,ONLY-LOCAL
2012-02-17 15:40:01

 Name      Value
 ----------------
 ThNetbTtl 38728
 ThNetbSnd 38728
 ThNetbRcv 0
 ThDiskTtl 16384
 ThDiskSnd 16384
 ThDiskRcv 0
 ThRqto    1
 ThSuct    0
 ThAbrt    0
 ThIntr    0
 ThUsrf    0
 ThCofl    0
 ThCobr    0
 StRqas    0.00
 StRqaa    8.66
 StRqwt    1.66
 StRqhd    0.00
 StRqsp    0.00
 StRqlk    0.00
 StCLim    16
 StCAct    37
 StRqLim   1000
 StRqAct   1
 StOftr    ON
 StFtmr    OFF
 StFtpr    OFF
```

Explanation of output:

The default output format begins with a header containing the following specifications:
– Name of the openFT instance and selected data format (*raw* or *formatted*)
– Monitoring start time and partner and request selection
– Current timestamp

This is followed by the list of default values, see also page 324.

6.40 ftshwo - Display operating parameters

The *ftshwo* command outputs the operating parameters of the local openFT system. Alongside the standard output and output in CSV format, output may also be specified as a platform-specific command sequence. In this way, it is possible to save the settings and then load them onto another computer with the selected operating system.

The FT administrator can set or modify the operating parameters with the *ftmodo* command.

The transfer admission of the ADM trap server is not output with the default output format and CSV output format. It only appears as a command sequence in the output (*-px*, *-pw*, *-p2*, *-pz*) .

Format

ftshwo -h |
[-csv | -px | -pw | -p2 | -pz]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-csv The operating parameters are output in CSV format. The individual values are separated by semicolons.

-px The operating parameters are output as a command string. This can be called as a shell procedure on Unix systems in order to regenerate the identical operating parameters.

-pw The operating parameters are output as a command string. This can be called as a batch procedure on Windows systems in order to regenerate the identical operating parameters.

-p2 The operating parameters are output as a command string. This can be called as an SDF procedure on BS2000/OSD systems in order to regenerate the identical operating parameters.

-pz The operating parameters are output as a command string. This can be called as a Clist procedure on z/OS systems in order to regenerate the identical operating parameters.

No option specified
The operating parameters are output in standard format.

6.40.1 Output format of ftshwo

Example

```
ftshwo
STARTED PROC-LIM CONN-LIM ADM-CLIM RQ-LIM MAX-RQ-LIFE TU-SIZE KEY-LEN CCS-NAME
  YES     NONE       16        8     2000       30      65535    2048  ISO88591
PTN-CHK DYN-PART SEC-LEV  FTAC-LOG FT-LOG ADM-LOG     USE TNS USE CMX ENC-MAND
  STD       ON   B-P-ATTR    ALL     ALL    ALL          NO      NO       NO
OPENFT-APPL      FTAM-APPL          FTP-PORT           ADM-PORT         ADM-CS
*STD             *STD               21                 11000              NO
ACTIVE           ACTIVE             ACTIVE             ACTIVE
HOST-NAME        IDENTIFICATION / LOCAL SYSTEM NAME
*NONE            mc011.mynet.local / $FJAM,MC011

DEL-LOG  ON  AT     RETPD    ADM-TRAP-SERVER
 OFF   DAILY 00:00    14     *NONE

TRAP: SS-STATE FT-STATE PART-STATE PART-UNREA RQ-STATE TRANS-SUCC TRANS-FAIL
CONS    OFF      OFF       OFF        OFF       OFF       OFF        OFF
ADM              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
```

Meaning of the output together with the associated command options:

Field name	Meaning and values	Command/ option
STARTED	Specifies whether the asynchronous openFT server has started (YES) or not (NO).	*ftstart* *ftstop*
PROC-LIM	Maximum number of openFT servers available for the processing of asynchronous requests.	*ftmodo -pl=*
CONN-LIM	Maximum number of asynchronous requests that can be processed simultaneously.	*ftmodo -cl=*
ADM-CLIM	Maximum number of asynchronous administration requests including ADM traps that can be processed simultaneously.	*ftmodo -admcl=*
RQ-LIM	Maximum number of file transfer requests that can simultaneously be present in the local system's request queue.	*ftmodo -rql=*
MAX-RQ-LIFE	Maximum lifetime of requests in the request queue (in days).	*ftmodo -rqt=*

Field name	Meaning and values	Command/ option
TU-SIZE	Upper limit for message length at transport level (in bytes).	*ftmodo -tu=*
KEY-LEN	Length of the RSA key currently used to encrypt the AES/DES key.	*ftmodo -kl=*
CCS-NAME	Name of the character set used by default for file transfer requests, see page 242	*ftmodo -ccs=*
PTN-CHK	Setting for sender verification: ADDR: address STD: identification	*ftmodo -ptc=*
DYN-PART	Setting for dynamic partner entries: ON (activated) OFF (deactivated)	*ftmodo -dp=*
SEC-LEV	Default security level for partners in the partner list for which no security level has been set:	*ftmodo -sl=*
	1..100: Fixed security level. 1 is the lowest and 100 the highest security level.	
	B-P-ATTR: The security level is depending on the partner's attributes, i.e.: 10 if the partner has been authenticated. 90 if the partner is known in the transport system. 100 otherwise, i.e. if the partner has only been identified by its address.	*ftmodo -sl=p*
FTAC-LOG	Scope of FTAC logging: ALL: All FTAC access checks MODIFY: Modifying file management requests and rejected FTAC access checks REJECTED: Only rejected FTAC access checks	*ftmodo -lc=*
FT-LOG	Scope of FT logging: ALL: All requests FAIL: Only errored FT requests NONE: FT Logging deactivated	*ftmodo -lt=*

Field name	Meaning and values	Command/ option
ADM-LOG	Scope of ADM logging: ALL: All requests FAIL: Only errored ADM requests MODIFY: only modifying ADM requests NONE: ADM Logging deactivated	*ftmodo -la=*
USE TNS	Specifies whether the TNS is to be used (YES) or not (NO) during operation with CMX	*ftmodo -tns=*
USE CMX	Specifies whether operation with CMX is activate (YES) or not (NO)	*ftmodo -cmx=*
ENC-MAND	Specifies whether inbound and/or outbound encryption is activated	*ftmodo -c=*
OPENFT-APPL	Port number of the local openFT server, possibly extended by the transport selector. *STD means that the default value is used i.e. 1100 and $FJAM in Transdata format (EBCDIC, 8 characters long, padded with blanks). Line 2: ACTIVE: openFT protocol activated DISABLED: openFT protocol (inbound) deactivated INACT: openFT protocol (inbound) not available	*ftmodo -openft=* *ftmodo -acta=*
FTAM-APPL	Port number of the local FTAM server, possibly extended by the transport selector, the session selector and the presentation selector. *STD means that the default value is used i.e. 4800 and $FTAM in Transdata format (EBCDIC, 8 characters long, padded with blanks) Line 2: ACTIVE: FTAM protocol activated DISABLED: FTAM protocol (inbound) deactivated INACT: FTAM protocol (inbound) not available NAVAIL: FTAM not installed	*ftmodo -ftam=* *ftmodo -acta=*
FTP-PORT	Port number used by local FTP server. Default port: 21	*ftmodo -ftp=*

Field name	Meaning and values	Command/ option
	Line 2: ACTIVE: FTP protocol activated DISABLED: FTP protocol (inbound) deactivated INACT: FTP protocol (inbound) not available NAVAIL: FTP not installed	*ftmodo -acta=*
ADM-PORT	Port number used by remote administration. Default port: 11000 Line 2: ACTIVE: remote administration activated DISABLED: remote administration (inbound) deactivated INACT: remote administration (inbound) not available	*ftmodo -adm=* *ftmodo -acta=*
ADM-CS	Specifies whether the local openFT instance is flagged as a remote administration server (YES) or not (NO).	*ftmodo -admcs=*
HOST-NAME	Host name of the local computer, *NONE means that no host name has been assigned.	*ftcrei -addr=* *ftmodi -addr=*
IDENTIFICATION	Instance identification of the local openFT instance.	*ftmodo -id=*
LOCAL-SYSTEM-NAME	Name of the local system.	*ftmodo -p= -l=*
DEL-LOG	Automatic deletion of log records activated (ON) or deactivated (OFF)	*ftmodo -ld=*
ON	Day on which the log records are to be deleted: MON, TUE, ... SUN (day of the week) or 1...31 (day of the month) or DAILY (every day)	*ftmodo -ldd=*
AT	Time at which the log records are to be deleted (hh:mm)	*ftmodo -ldt=*
RETPD	Minimum age of log records for deletion in days. 0 means the current day.	*ftmodo -lda=*
ADM-TRAP-SERVER	Name or address of the partner to which the ADM traps are sent. *NONE means that the sending of ADM traps is deactivated.	*ftmodo -atpsv=*

Field name	Meaning and values	Command/ option
TRAP	The TRAP settings are output here. The possible values are ON and OFF. The row CONS indicates the console traps and the row ADM the ADM traps. The columns designate the events for which traps may be generated: SS-STATE: Change of the status of the openFT subsystem (row CONS only) FT-STATE: Change of the status of the asynchronous server PART-STATE: Change of the status of partner systems PART-UNREA: Partner systems unreachable RQ-STATE: Change of the status of request administration TRANS-SUCC Requests completed successfully TRANS-FAIL: Failed requests	*ftmodo* *-tpc=* *-atp=*
FUNCT	The settings for monitoring (MONITOR row) and tracing (TRACE row) are output in this section. The individual columns have the following meanings: SWITCH: Function (monitoring or tracing) activated (ON) or deactivated (OFF) PARTNER-SELECTION: Selection based on the partner system's protocol type. Possible protocol types: OPENFT, FTP, FTAM. ADM (administration partner) can also be output under TRACE. ALL means that all protocol types have been selected, i.e. tracing/monitoring is possible for all partner systems. NONE means that no protocol type has been selected.	*ftmodo* *-mon=* *-tr=* *ftmodo* *-monp=* *-trp=*

Field name	Meaning and values	Command/ option
FUNCT (*cont.*)	REQUEST-SELECTION: Selection based on the request type. The following are possible: ONLY-SYNC/ONLY-ASYNC (only synchronous or only asynchronous requests) ONLY-LOCAL/ONLY-REMOTE (only locally or only remotely submitted requests). ALL means no restriction, i.e. all requests.	*ftmodo* *-monr=* *-trr=*
	OPTIONS (only in the TRACE row) NONE means no options (trace in default format) NO-BULK-DATA means minimum trace, i.e. bulk data (file contents) is not logged. In addition, no repetitions of data log elements are logged.	*ftmodo -tro=*
	OPTIONS-LL Scope of tracing for lower protocol layers: OFF: Deactivated STD: Default DETAIL: Details	*ftmodo -troll=*

6.41 ftshwp - Display FT profiles

ftshwp stands for "show profile" and allows you to obtain information about FT profiles. In short form, it displays the names of the selected FT profiles, as well as the following information:

– whether or not the FT profile is privileged: asterisk (*) before the profile name
– whether or not the transfer admission is disabled: exclamation mark (!) before the profile name.

As the ADM administrator, you may also obtain information about ADM profiles (i.e. FT profiles with the property "access to remote administration server").

As the FTAC administrator, you may obtain information about all FT profiles in the system.

Format

ftshwp -h |
[<profile name 1..8> | @s]
[-s=[<transfer admission 8..32> | @a | @n]
[,<user ID 1..32> | @a | @adm]]
[-l][-csv]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

profile name | **@s**
Is the name of the FT profile you wish to see.

@s for *profile name*
Provides information on the standard admission profile for the user ID if this has been set up. Otherwise you see a corresponding message.

profile name not specified
Profile name is not used as a criterion for selecting the FT profile to be displayed. If you do not specify the profile with -s (see below), FTAC will display information on all of your FT profiles.

-s=[transfer admission | **@a** | **@n**][**,**user ID | **@a**]
-s is used to specify criteria for selecting the FT profiles to be displayed.

If you wish to view standard admission profile, you can only specify *@n* or *@a*.

Transfer admission
Is the transfer admission of the FT profile to be displayed. A binary transfer admission must be specified in hexadecimal format in the form x\'...\' or X\'...\' .

@a for *transfer admission*

Displays information either on the FT profile specified with *profile name* (see above) or (if no *profile name* was specified) on all FT profiles.

As the FTAC administrator, you can specify *@a* if you want to obtain information on FT profiles belonging to other login names, since even you should not know the transfer admission.

@n for *transfer admission*

displays information on FT profiles that do not have a defined transfer admission.

As the FTAC administrator, you can specify *@n* if you want to obtain information on FT profiles belonging to other login names which do not have a defined transfer admission.

transfer admission not specified

causes FTAC to query the transfer admission on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the transfer admission. To exclude the possibility of typing errors, the program prompts you to enter the transfer admission a second time. If you just press <ENTER>, this has the same effect as specifying *@a*.

,user ID

As the FTAC administrator, you can specify any login name here.

@a for *user ID*

As the FTAC administrator, you can obtain information on the FT profiles of all login names.

As the ADM administrator, you can obtain information on the own FT profiles and the ADM profiles.

@adm for *user ID*

As the FTAC or ADM administrator, you obtain information on ADM profiles.

user ID not specified

displays only profiles belonging to the user's own login name, regardless of who issues the command.

-s not specified

if no profile name is specified, displays all the FT profiles belonging to the login name under which the *ftshwp* command is issued. Otherwise, displays information on the FT profile with the specified name.

-l displays the contents of the selected FT profiles.

In long form, the entire contents of the selected FT profiles are displayed. The USER-ADM parameter contains the following information:
– the login name for which an admission profile is valid or if it is an ADM profile
– whether or not it is valid for a specific password of the login name
– whether or not it is valid for any password of the login name
– whether or not it has an undefined password and is thus disabled.

Please note that ADM profiles always are indicated by the value *ADM under the USER-ADM parameter.

USER-ADM=	**Meaning**
(user ID,,OWN)	Profile is valid for all passwords of the login name.
(user ID,,YES)	The profile is valid only for a specific password of the login name (specified in *-ua=user ID, password* with an *ftcrep* or *ftmodp* command). The profile is deactivated (not disabled) if the password is changed. You can activate it again, for example, by resetting the password.
(user ID,, NOT-SPECIFIED)	The FTAC administrator created or modified the FT profile knowing only the login name. As a result, the profile was disabled. You must enable the profile with *ftmodp* and the *-v=y* parameter.

If an FT profile is disabled, the *TRANS-ADM* parameter indicates the reasons why the profile was disabled. The following table shows the possible parameter values, as well as their meanings:

TRANS-ADM=	**Possible cause and action**
NOT-SPECIFIED	The FTAC administrator created the FT profile without transfer admission, or the FTAC user did not specify transfer admission. Measure: specify transfer admission
DUPLICATED	An attempt was made to create an FT profile with the same transfer admission. Measure: specify new transfer admission
LOCKED (by_adm)	The FTAC administrator modified the FT profile by login name only. The transfer admission remained unchanged but was disabled. Measure: enable the profile using the *ftmodp* command and the *-v=y* parameter
LOCKED (by_import)	The FT profile was created using the *ftimpe* command. The transfer admission remains unchanged, but is marked as disabled. Measure: enable the profile using the *ftmodp* command and the *-v=y* parameter.

TRANS-ADM=	Possible cause and action
LOCKED (by_user)	The FTAC user disabled his/her own FT profile. Measure: enable profile using the *ftmodp* command and the *-v=y* parameter.
EXPIRED	The time up to which the transfer admission can be used has expired. Measure: enable profile using the *ftmodp* command and the *-d* parameter, by removing the temporal restriction using the *-d* entry and defining a new time span with *-d=date*.

ftshwp does not provide a means of displaying a transfer admission. If you have forgotten a transfer admission, you have to define a new one using *ftmodp*.

-l not specified
: displays only the names of your FT profiles. Markings also indicate whether or not an FT profile is privileged (*) and whether or not it is disabled (!).

-csv You can use *-csv* to specify that the FT profiles are to be output in the CSV format. The values in the output are separated by semicolons. If *-csv* is specified, output is always in long form (analogous to *-l*) regardless of whether or not *-l* has also been specified.

-csv not specified
: The FT profiles are output in the standard format, i.e. in abbreviated form if *-l* is not specified and in detailed form if *-l* is specified.

Examples

1. You are an FTAC administrator and want to view all the standard admission profiles on your system.

   ```
   ftshwp @s -s=@n,@a -l
   ```

   Output takes the following form:

   ```
   *STD
    TRANS-ADM   = (NOT-SPECIFIED)
    USER-ADM    = (john,,OWN)
    FT-FUNCTION = (TRANSFER-FILE, MODIFY-FILE-ATTRIBUTES, READ-FILE-
   DIRECTORY)
    LAST-MODIF  = 2012-03-23 17:12:25
   *STD
    TRANS-ADM   = (NOT-SPECIFIED)
    WRITE       = NEW-FILE
    USER-ADM    = (dagobert,,OWN)
    FT-FUNCTION = (TRANSFER-FILE)
    LAST-MODIF  = 2012-03-22 16:06:55
   ```

2. You are the FT administrator and wish to view the profile *acctrap1* on the ADM trap server.

   ```
   ftshwp acctrap1 -l
   ```

   Output takes the following form:

   ```
   acctrap1
    USER-ADM    = (ADMIN002,,OWN)
    FT-FUNCTION = (ADM-TRAP-LOG)
    LAST-MODIF  = 2012-01-23 18:24:42
   ```

   The value ADM-TRAP-LOG under FT-FUNCTION in the *acctrap1* profile means that the remote administration server can receive ADM traps with this profile.

3. You are the ADM administrator and wish to view the ADM profiles on the remote administration server.

   ```
   ftshwp -s=@a,@adm  -l
   ```

   Output takes the following form:

   ```
   acccentr
    USER-ADM    = (*ADM,,OWN)
    FT-FUNCTION = (ACCESS-TO-ADMINISTRATION)
    LAST-MODIF  = 2012-01-23 18:21:08
   ```

   The profile *acccentr* is a ADM profile. This is indicated by the value ACCESS-TO-ADMINISTRATION under FT-FUNCTION and the value *ADM for user ID under USER-ADM.

4. You are the FT administrator and would like to view the profile *remadmin* that has been set up for remote administration.

   ```
   ftshwp remadmin -l
   ```

   Output takes the following form:

   ```
   remadmin
    USER-ADM    = (ADMIN001,,OWN)
    FT-FUNCTION = (REMOTE-ADMINISTRATION)
    LAST-MODIF  = 2012-02-27 16:20:38
   ```

6.42 ftshwptn - Display partner properties

You use the *ftshwptn* command to call up the following information about the partner systems entered in the partner list:

- The name of the partner system
- The status of the partner system (activated, deactivated)
- The security level that was assigned to the partner system
- The priority that was assigned to the partner system
- The setting for the openFT trace function for the partner system
- The number of file transfer requests to the partner system issued in the local system that have not yet been completed
- The number of file transfer requests for the local system that have been issued in the partner system
- The mode for sender verification and authentication
- The partner system's transport address, possibly with the port number if this is different from the default
- The identification of the partner system
- The routing information if the partner system can only be accessed via an intermediate instance

You can also output the partners in the partner list as a platform-specific command sequence. In this way, it is possible to save the partner list and load it at another computer which may possibly be running a different operating system.

Format

```
ftshwptn  -h |
          [ <partner 1..200> | @a ]
          [ -st=a | -st=na | -st=d | -st=ie | -st=nc | -st=ad | -st=da ]
          [ -l | -csv | -px | -pw | -p2 | -pz ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

partner | **@a**
Specifies the partner whose properties you want to display. You can specify the name of the partner in the partner list or the address of the partner system. For details in address specifications, see section “Specifying partner addresses” on page 68

@a for *partner*
The properties of all the partners in the partner list are displayed.

partner not specified
The properties of all the partners in the partner list are displayed.

-st=a | **-st=na** | **-st=d** | **-st=ie** | **-st=nc** | **-st=ad** | **-st=da**
This operand enables you to display the properties of partner systems which have a specific status. You an specify the following values:

a (active)
All the partner systems with the status ACTIVE are displayed.

na (not active)
All the partner systems which do **not** have the status ACTIVE are displayed.

d (deactivated)
All the partner systems with the status DEACTIVE are displayed.

ie (installation error)
All the partner systems with the status LUNK, RUNK, LAUTH, RAUTH, NOKEY or IDREJ are displayed.

nc (not connected)
All the partner systems with the status NOCON or DIERR are displayed.

ad (active + automatic deactivation)
All the partner systems for which the option AUTOMATIC-DEACTIVATION is set (see the option *-ad* in the *ftaddptn* and *ftmodptn* commands) but are still active are displayed.

da (deactivated + automatic deactivation)
All the partner systems which have actually been deactivated because of the AUTOMATIC-DEACTIVATION option are displayed.

-st not specified
The output is not restricted to partner systems with a specific status.

-l | -csv | -px | -pw | -p2 | -pz

These options determine the scope and format of the output.

-l The properties of the partner systems are output in full as a table.

-csv The properties of the partner systems are output in CSV format. The individual values are separated by semicolons.

-px The properties of the partner systems are output as a command sequence. This can be called in Unix systems as a shell procedure in order to generate partner entries with identical properties.

-pw The properties of the partner systems are output as a command sequence. This can be called in Windows systems as a batch procedure in order to generate partner entries with identical properties.

-p2 The properties of the partner systems are output as a command sequence. This can be called in BS2000 systems as an SDF procedure in order to generate partner entries with identical properties.

-pz The properties of the partner systems are output as a command sequence. This can be called in z/OS systems as a CLIST procedure in order to generate partner entries with identical properties.

-l, *-csv*, *-px*, *-pw*, *-p2*, *-pz* not specified

If you do not specify any of these options then the partners' properties are output in their abbreviated form.

6.42.1 Output format of ftshwptn

Example for the output in abbreviated form and in full format:

```
ftshwptn

NAME     STATE SECLEV   PRI  TRACE  LOC  REM P-CHK ADDRESS
pingftam ACT   50       NORM FTOPT    0    0       ftam://PING.homenet.de
PINGO    ACT   STD      NORM FTOPT    0    0 FTOPT PINGPONG.homenet.de:1234
rout0001 ACT   STD      HIGH FTOPT    0    0 FTOPT INCOGNITO
servftp  ACT   B-P-ATTR LOW  ON       0    0       ftp://ftp.homenet.de

ftshwptn -l

NAME     STATE SECLEV   PRI  TRACE  LOC  REM P-CHK ADDRESS
         INBND REQU-P                             ROUTING  IDENTIFICATION
pingftam ACT    50      NORM FTOPT    0    0       ftam://PING.homenet.de
         DEACT STD
PINGO    ACT   STD      NORM FTOPT    0    0 FTOPT PINGPONG.homenet.de:1234
         ACT   SERIAL                                      PINGPONG.homenet.de
rout0001 ACT   STD      HIGH FTOPT    0    0 FTOPT INCOGNITO
         ACT   STD                                ROUT01   INCOGNITO.id.new
servftp  ACT   B-P-ATTR LOW  ON       0    0       ftp://ftp.homenet.de
         ACT   STD
```

Explanation

NAME
: Name of the entry in the partner list.

STATE
: Specifies how file transfer requests issued locally to the specified partner system are processed.

  ACT
  : File transfer requests issued locally to this partner system are processed with *ftstart*.

  DEACT
  : File transfer requests issued locally to this partner system are initially not processed, but are only placed in the request queue.

  ADEAC
  : Failed attempts at establishing a connection lead to this partner system being deactivated. The maximum number of consecutive failed attempts is 5. In order to perform file transfers with this partner system again, it must be explicitly reactivated with *ftmodptn -st=a*.

  NOCON
  : Attempt to establish a transport connection failed.

LUNK
: Local system is not known in the remote FT system.

RUNK
: Partner system is not known in the local transport system.

AINAC
: Partner system has been deactivated after a number of unsuccessful attempts to establish a connection.

LAUTH
: Local system could not be authenticated in the partner system. A valid public key for the local openFT instance must be made available to the partner system.

RAUTH
: Partner system could not be authenticated in the local system. A valid public key for the partner system must be stored in the folder *syskey* of the openFT instance, see also "Instance directory" on page 26. In the case of the default instance, *syskey* is in the directory */var/openFT/std*.

DIERR
: A data integrity error has been detected on the connection to the partner system. This can be the result of attempts at manipulation on the data transfer path or of an error in the transport system. The connection has been interrupted, but the affected request is still live (if it has the capability of being restarted).

NOKEY
: The partner does not accept unencrypted connections, but no key is available in the local system. A new key must be generated.

IDREJ
: The partner or an intermediate instance has not accepted the instance ID sent by the local system. Check whether the local instance ID matches the entry for the partner in the partner list.

SHORT
: A resource bottleneck has occurred on the partner.

SECLEV
: Security level assigned to the partner system.

1..100
: A fixed security level is assigned to the partner system:
1 is the lowest security level (partner is extremely trusted) and 100 is the highest security level (partner is not trusted).

STD
: The global setting for the security level applies.

B-P-ATTR
: The security level is assigned to the partner on the basis of the partner's attributes, i.e.:
  – Security level 10 if the partner has been authenticated.
  – Security level 90 if the partner is known in the transport system and is identified by the name it is known by in the transport system.
  – Security level 100 otherwise, i.e. if the partner has only been identified by its address.

PRI
: Priority of a partner with respect to the processing of requests:

  NORM
  : Normal priority.

  LOW
  : Low priority.

  HIGH
  : High priority.

TRACE
: The global settings for partner selection in the openFT trace function apply.

  FTOPT
  : The global setting for partner selection in the openFT trace function applies.

  ON
  : The trace function is activated for this partner. However, a trace is only written if the global openFT trace function is also activated. For details, see section “Activating partner specific trace” on page 378.

  OFF
  : The trace function is deactivated for this partner.

LOC
: Shows the number of file transfer requests addressed to the partner system entered in the local system.

REM
: Shows the number of file transfer requests issued by the remote FT system and addressed to the local FT system.

P-CHK
: Shows the settings for sender verification and authentication.

  FTOPT
  : The global setting for sender verification applies.

  STD
  : Checking of the transport address is deactivated. Only the identification of the partner is checked. The transport address of the partner is not checked even if extended sender verification is activated globally.

  T-A
  : Checking of the transport address is activated. The transport address of the partner is checked even if checking of the transport address is deactivated globally. If the transport address used by the partner to log in does not correspond to the entry in the partner list, the request is rejected.

AUTH
: The partner is subjected to a cryptographic identity check on the basis of its public key in the *syskey* directory (for authentication). The partner supports authentication level 2.

!AUTH
: The partner is subjected to a cryptographic identity check on the basis of its public key in the *syskey* directory (for authentication). The partner supports authentication level 1.

AUTHM
: Authentication must be used.

NOKEY
: No valid key is available from the partner system although authentication is required.

ADDRESS
: Address of the partner system.

ROUTING
: Routing info of the partner system if specified. The routing info is only output with *ftshwptn* -l.

IDENTIFICATION
: Identification of the partner system if specified. The identification is only output with *ftshwptn -l*.

INBND State of the partner for inbound requests:

ACT Inbound function is activated, i.e. requests issued remotely are processed.

DEACT
: Inbound function is deactivated, i.e. requests issued remotely are rejected.

REQU-P Operating mode for asynchronous outbound requests:

STD Requests to this partner can be processed in parallel.

SERIAL
: Requests to this partner are always processed serially.

6.43 ftshwr - Display request properties and status

The *ftshwr* ("show requests") command allows you to request information about FT requests. You can specify selection criteria in order to obtain information about specific FT requests.

The FT administrator can obtain information about the requests of any owner.

Format

ftshwr -h |
[-ua=<user ID 1..32> | -ua=@a]
[-ini=l | -ini=r | -ini=lr | -ini=rl]
[-st=a | -st=w | -st=l | -st=c | -st=f | -st=h | st=s]
[-pn=<partner 1..200>]
[-fn=<file name 1..512>]
[-gid=<global request identification 1..4294967295>]
[-s | -l][-csv]
[<request ID 1..2147483647>]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-ua=user ID | **-ua=@a**
You use *-ua* to specify the user ID for which requests are to be displayed.

user ID
As a user, you can only specify your own user ID.
As an FT administrator, you may specify any user ID here.

@a As an FT administrator, you can specify @a to display requests for all user IDs.

-ua= not specified
Your own user ID is the selection criterion.
Exception: The FT administrator has called the command and also specified a request ID: in this case, the presetting is *@a*.

-ini=l | -ini=r | -ini=lr | -ini=rl

You use *-ini* to specify the initiator for which you want to display requests. The following specifications are possible:

l (local) Only locally submitted requests are displayed.

r (remote) Only remotely submitted requests are displayed.

lr, rl (local + remote) Both locally and remotely submitted requests are displayed.

-ini not specified

The initiator is not the selection criterion (corresponds to *lr* or *rl*).

-st=a | -st=w | -st=l | -st=c | -st=f | -st=h | -st=s

If you specify *-st* then only information on requests with the corresponding status is output.

The following specifications are possible:

a (active)
The request is currently being executed.

w (wait)
The request is waiting to be executed.

l (locked)
The request is locked.

c (cancelled)
The request has been deleted.

f (finished)
The request has already been executed.

h (hold)
The starting time specified on the issue of the request has not yet been reached.

s (suspend)
The request was interrupted, i.e. it is currently in the SUSPEND status.

-pn=partner

You use *-pn* to specify a name or an address for the partner system for which you want to display requests. The partner should be specified as on request submission or as output by the *ftshwr* command without the *-s*, *-l* or *-csv* option. If openFT finds a partner in the partner list for a specified partner address then *ftshwr* displays the name of the partner even if a partner address was specified at the time the request was entered.

-fn=file name

You use *-fn* to specify the file name for which requests are to be displayed. Requests that access this file in the local system are displayed.

You must specify the file name that was used when the request was issued. This file name is also output by the *ftshwr* command without the *-fn* option.

Wildcards are not permitted in the file name.

-gid=global request identification

With *-gid*, you specify the global request ID for a specific request that is to be displayed. The global request ID is only relevant for inbound requests from openFT and FTAM partners. It is assigned by the initiator of the request (transfer ID) and is sent to the local system.

-gid= not specified

The global request ID is not used as a selection criterion.

-s (sum) specifies that a summary overview of requests is to be output. For each possible request status (see the *-st* option), this overview indicates the number of requests that have this status.

-l (long form) specifies that the request properties are to be output in full.

-csv Specifies that the request properties are to be output in CSV format. If you also specify *-s* then the summary overview is output in CSV format. The values in the overview are output separated by semicolons.

-s, *-l* and *-csv* not specified

The request attributes are output in standard form.

request ID

request ID specifies the identification of a specific request that is to be output. The request ID is output on the screen on acknowledgment of receipt of the request. It can also be viewed, for example, using the *ftshwr -l* command.

If you have specified a request ID and the other specified criteria do not correspond to the request then the request is not displayed and the following error message is output:

```
ftshwr: Request request ID not found
```

6.43.1 Output format of ftshwr

6.43.1.1 Standard ftshwr output

```
$ftshwr
TRANS-ID   INI STATE PARTNER  DIR  BYTE-COUNT  FILE-NAME
65558      LOC WAIT  *PINGO   TO   0           /home1/september.pdf
196610     LOC WAIT  servus.* FROM 0           /home2/mails/memo02.txt
262146     LOC WAIT  servus.* TO   0           /home3/pic/picture10.gif
```

Description of the output

TRANS-ID
: The TRANS-ID column (transfer identification) contains the request numbers used by openFT to identify the file transfer requests. The TRANS-ID can be used to cancel requests with the *ftcanr* command.

INI
: The INI column indicates the initiator:
LOC: The request was submitted in the local system.
REM: The request was submitted in the remote system.

STATE
: The STATE column indicates the priority of the request.

  The priority is displayed after the state identifier. The only possible display is *l* for "low". If the request has the priority *normal* then nothing is displayed.

  The following states are possible:

  ACT (active)
  : The request is currently being processed.

  WAIT (wait)
  : The request is waiting.
  In this case, the partner system (PARTNER) may be indicated. This indication shows the cause of the *WAIT* state.

  LOCK (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.
  In this case, the partner system (PARTNER) may be indicated. This indication shows the cause of the *LOCKED* state.

CANC (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.

FIN (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 (hold)
: The start time specified when the request was issued has not been reached.

SUSP (suspend)
: The request was interrupted.

PARTNER
: Name or address of the partner, see also page 68. If the partner address is more than 8 characters in length then it is truncated to 7 characters and suffixed with an asterisk (*).

If the request is in a WAIT or LOCKED state, the following indicators before PARTNER are also entered in the request queue:

␣ (empty) No resources free at present (e.g. no memory).

* The local FT administrator has locked the resource, e.g. deactivating the partner.

! Connection setup to the partner system failed. The partner is currently inactive, or it can currently accept no further connections, or a network node has crashed.
Other possibilities: The connection to the partner system has been lost; a data integrity error has been detected.

? An installation or configuration error has occurred (e.g. the local system is not known to the partner), authentication of one of the partners has failed, or the encryption is local, or not available to the partner system.

DIR
: The DIR column specifies the direction of transfer.

TO Send to the remote system.

FROM
: Fetch from the remote system.

BYTE-COUNT
: This column indicates the number of bytes transferred and saved up to now. The BYTE-COUNT counter is only updated at certain intervals.

FILE-NAME
: Name of the file in the local system.

6.43.1.2 Totaled ftshwr output

In the case of totaled output, a table showing the number of requests in the various request states is output (refer to the *State* column under the standard output for the meanings of the states):

```
ftshwr -s
 ACT   WAIT   LOCK   SUSP   HOLD    FIN      TOTAL
   3      2      0      0      0      0          5
```

6.43.1.3 Detailed output from ftshwr

Example for the detailed output of the request with request ID 131074:

```
ftshwr -l 131074

TRANSFER-ID =131074      STORE  =12-05-29 11:45:27  FILESIZE=514610
   STATE    =WAIT        BYTECNT=0
   INITIATOR=LOCAL       TRANS  =FROM               PRIO    =NORM
   WRITE    =REPLACE     START  =SOON               CANCEL  =NO
   COMPRESS =NONE        DATA   =CHAR
   TRANSP   =NO          ENCRYPT=NO
   TARGFORM =BLOCK       TRECFRM=STD
   OWNER    =maier       DICHECK=NO                RECFORM =VARIABLE
   PARTNER  =ftserv01.mycompany.net
   PARTNER-STATE = ACT
   PARTNER-PRIO  = NORM
   LOC: FILE     =/home2/memo02.txt
        TRANS-ADM=(maier)
        CCSN     =ISO88591
   REM: FILE     =/home/save/memo02.txt
        TRANS-ADM=(servelog)
```

Example of detailed output of inbound request with request ID 524410:

```
ftshwr -l 524410

TRANSFER-ID =524410      STORE  =12-06-14 14:33:24  FILESIZE=10485760
   STATE    =ACTIVE      BYTECNT=0                  RECSIZE =1024
   INITIATOR=REMOTE      TRANS  =FROM               PRIO    =
   WRITE    =REPLACE     START  =SOON               CANCEL  =NO
   COMPRESS =NONE        DATA   =CHAR               GLOB-ID =852520
   TRANSP   =NO          ENCRYPT=NO                 TABEXP  =NO
   OWNER    =user1       DICHECK=NO                 RECFORM =VARIABLE
   PARTNER  =ftserv.mycompany.net
   PARTNER-STATE =ACT
   PARTNER-PRIO  =NORM
   FILE     =par.file.S3.C31
   TRANS-ADM=(serv,)
```

Description of the output

TRANSFER-ID (transfer identification)
: Request ID which openFT uses to identify file transfer requests. Requests can be canceled using the *ftcanr* and the request ID.

STATE
: State of the request. Possible values:

  ACTIVE
  : The request is currently being processed.

  WAIT
  : The request is waiting. If the cause of the WAIT state is known, further information is indicated in the PARTNER-STATE field.

  LOCKED
  : The request is temporarily excluded from processing. This status can also occur at openFT and at FTAM partners.
  With openFT partners, when a resource bottleneck is encountered or if external data media must first be made available for example.
  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.
  If the cause of the LOCKED state is known, further information is indicated in the PARTNER-STATE field.

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

FINISHED
: This status occurs for requests involving FTAM partners when the request has either been completed or cancelled, but the user has not yet been informed of this.

HOLD
: The start time specified when the request was issued has not yet been reached.

SUSPENDED
: The request was interrupted.

INITIATOR
: This specifies where the request was issued. Possible values:

  LOCAL
  : The request was issued in the local system.

  REMOTE
  : The request was issued in the remote system.

WRITE
: This specifies whether the destination file is to be overwritten, extended or created. Possible values:

  OVERWRITE (default value)
  : If the destination file already exists, it is overwritten; otherwise, it is created.

  EXTEND
  : If the destination file already exists, the file sent is appended to the destination file; otherwise. If the destination file did not exist, it is created.

  NEW
  : A new destination file is created and written.

COMPRESS
: This specifies whether the file should be transferred with data compression.

  Possible values: BYTE, ZIP, NONE.

TRANSP
: Indicated whether the file is to be sent in transparent file format. Possible values: YES, NO

TARGFORM
: Format of the file in the target system.

  Possible values:

  STD (default value)
  : The file is saved in the same format as in the sending system.

  BLOCK
  : The file is saved in block format.

  SEQ
  : The file is saved as a sequential file.

OWNER
: Local login name.

PARTNER
: Name or address of the partner, see also page 68.

PARTNER-STATE
: Status of the partner. Possible values:

  ACT Activated

  DEACT
  : Deactivated

  NOCON
  : No connection, for example because the openFT server has not been started in the remote system.

  INSTERR
  : An installation or configuration error has occurred (the local system is not known to the partner, for instance), authentication of one of the partners has failed, or the encryption is local, or not available to the partner system.

  SHORT
  : A resource bottleneck has occurred on the partner.

PARTNER-PRIO
: Prioritization of the partner when processing requests.
  Possible values:

  LOW The partner has low priority.

  NORM
  : The partner has normal priority.

  HIGH
  : The partner has high priority.

LOC Properties in the local system:

FILE File name in the local system

TRANS-ADM
Transfer admission for the local system

CCSN
CCS name used in the local system. The CCSN is only output for text files.

SUCC-PROC
Local follow-up processing commands if successful (if specified in the request).

FAIL-PROC
Local follow-up processing commands if unsuccessful (if specified in the request).

REM Properties in the remote system:

FILE File name in the remote system

TRANS-ADM
Transfer admission in the remote system. Possible values:

REMOTE-PROFILE
request with FTAC transfer admission

TRANS-ADM=(*user ID*)
request with *user ID,,password*

CCSN
CCS name used in the remote system

SUCC-PROC
Remote follow-up processing commands if successful (if specified in the request).

FAIL-PROC
Remote follow-up processing commands if unsuccessful (if specified in the request).

STORE
Indicates the time at which the request was entered in the request queue.

BYTECNT
This value is output only if the request is currently active or if it was already active and the file transfer has been interrupted. BYTECNT indicates the number of bytes transferred and saved up to now. The counter is updated regularly.

TRANS
: This shows the direction of transfer. Possible values are:

  TO The document is sent.

  FROM The document is received.

START
: Indicates the time at which the request is to be started. Possible values:

  Date / Time
  : The date and time at which the request is to be started is output.

  SOON
  : The request should be started as soon as possible.

  No entry
  : The request was issued in the remote system.

DATA
: Indicates the file type. Possible values:

  CHAR (default value for openFT partners)
  : The file contains text with variable record lengths.

  BIN The file contains an unstructured sequence of binary data.

  USER
  : The file contains structured binary data with variable record length.

ENCRYPT
: Indicates whether data encryption was specified.
  Possible values: NO, YES.

TRECFRM
: Record format of the file in the target system

  Possible values:

  STD (default value)
  : The file is saved with the same record format as in the sending system.

  UNDEFINED
  : The file is saved with an undefined record format.

DICHECK
: Specifies whether the integrity of the data is to be checked.
  Possible values: NO, YES.

FILESIZE
: Size of the file in bytes. If the output is followed by a "K", the output is in kilobytes. If it is followed by an "M", the output is in megabytes. The size is indicated here only if the request was already active. For receive requests, a value is indicated here only if the partner has sent one with the request.

PRIO Request priority. Possible values:

NORM
: The request has normal priority

LOW
: The request has low priority

No entry
: The request was issued in the remote system.

CANCEL
: If the "Cancel-Timer" was set, the time at which the request is deleted from the request queue is indicated here. If no cancel time was specified in the request, NO is output.

GLOB-ID
: Global request identification, displayed only in the case of inbound requests from openFT and FTAM partners (INITIATOR=REMOTE). This corresponds to the request identification (=TRANSFER-ID) on the initiator system.

RECFORM
: Record format.
Possible values: UNDEFINED, VARIABLE, FIX.

RECSIZE
: Maximum record size, if specified.

DIAGCODE
: This column is usually empty. Otherwise, it provides further diagnostic information on operational states in the form of a CMX return code or an FTAM or openFT diagnostic code. FTNEA diagnostic codes have the format NEBFnnnn (NEABF) or NEBDnnnn (NEABD). The following openFT diagnostic codes have been defined:

Value	Meaning
0	No cause specified.
1	Connection setup normal.
2	There is a resource bottleneck.
3	There is a resource bottleneck; the connection will be set up later by the rejecting entity.
4	Initialization is not yet complete.

Value	Meaning
5	SHUTDOWN is in progress.
6	The requesting entity is unknown.
7	A protocol error has occurred.
8	A transport error has occurred.
9	A system error has occurred.
10	This code is reserved (for SN77309 part 5).
11	The connection is not accepted without encryption.

FTAM diagnostic codes have the format `FTAMnnnn` and values from the ISO 8571-3 standard. An extract of possible diagnostic codes taken from the standard can be found in the section of the same name in the User Guide.

The following values are only output for FTAM partners:

STOR-ACCOUNT
: Account number; is output only if specified by the user.

AVAILABILITY
: Possible values: IMMEDIATE, DEFERRED.
Is output only if specified by the user.

ACCESS-RIGHTS
: Access mode

Possible values: combinations of r, i, p, x, e, a, c, d.
Is output only if specified by the user.

LEGAL-QUAL
: Legal qualification

Is output only if the local system is the initiator and the value is specified by the user.

6.44 ftstart - Start asynchronous openFT server

This command starts the asynchronous openFT server. This processes all the requests stored in the request queue as well as all the inbound requests.

When the asynchronous openFT server is started, the protection bit settings for files that are created on inbound requests are set implicitly. The settings for the shell under which you entered *ftstart* apply. For more details, see section “Setting the protection bit for newly created files” on page 59.

It is necessary to shut down and restart the asynchronous openFT server, for example, if you want to switch between operation with and without CMX.In the case of openFT on Solaris, please refer to section “Solaris SMF” on page 41.

Format

ftstart [-h]

Description

-h Displays the command syntax on the screen.

6.45 ftstop - Stop asynchronous openFT server

This command shuts down the asynchronous openFT server. After this, no further inbound requests and no locally submitted asynchronous requests are processed:

– Inbound requests are rejected

– Locally submitted asynchronous requests are stored in the request queue

Once the *ftstop* command has been issued, the asynchronous openFT server is not stopped until all the server processes have been terminated. This may take a few minutes if, for example, disconnection is delayed due to line problems.

When the asynchronous openFT server is restarted, the requests present in the request queue are processed normally. Requests that were cancelled due to the shutdown of the asynchronous openFT server are relaunched provided that the partner supports this function.

In the case of openFT on Solaris, please refer to section “Solaris SMF” on page 41.

Format

ftstop [-h]

Description

-h Displays the command syntax on the screen.

6.46 ftupdi - Update the instance directory

Using *ftupdi*, you can update an instance file tree that was made using openFT V10.0 or V11.0 so that it can continue to be used with openFT V12.0. The settings of the operating parameters, FTAC admission sets, FTAC admission profiles and log records are retained.

Any requests for this instance which are still present will be lost.

Format

ftupdi -h | <directory 1..128>

Description

-h Displays the command syntax on the screen. Any entries after *-h* are ignored.

directory
Here, you enter the directory which contains the instance file tree of the instance to be updated.

Messages of the ftupdi command

If *ftupdi* could not be carried out as specified, an explanatory message is displayed; the exit code will then be "not equal to zero".

Example

The FT administrator wants to update the directory of the instance *hugo*.

```
ftupdi /var/openFT/.hugo
```

6.47 ftupdk - Update public keys

Using *ftupdk*, you can update the public key files of existing key pair sets.

For example, you can use it to insert updated comments from the *syspkf.comment* file into existing public key files or replace accidentally deleted public key files of a key pair set.

Format

ftupdk [-h]

Description

-h Displays the command syntax on the screen.

Example

The name of the FT administrator is to be imported into the public key files. First, the file *syspkf.comment* is edited using an editor. This file is located in the *config* subdirectory of the instance directory, see the *ftcrei* command on page 184.

The file might, for example, contain only the following line:

```
FT administrator: John Smith, Tel. 12345
```

The command is:

```
ftupdk
```

The command is executed without an error message. Following this, the information will be placed at the beginning of all *syspkf...* public key files as a comment line.

6.48 install.ftam - Install openFT-FTAM

The *install.ftam* command allows you to install and uninstall openFT-FTAM. Installation is only permitted if you have an openFT-FTAM license.

The *install.ftam* command is located in the */opt/openFT/bin/ftbin* directory.

Format

install.ftam -h | -i | -d

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-i openFT-FTAM is installed.

-d openFT-FTAM is uninstalled.

6.49 install.ftp - Install openFT-FTP

You use the *install.ftp* command to install and uninstall openFT-FTP. Installation is only permitted if you have an openFT-FTP license.

The *install.ftp* command is located in the */opt/openFT/bin/ftbin* directory.

Format

install.ftp -h | -i | -d

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-i openFT-FTP is installed.

-d openFT-FTP is uninstalled.

[1] - stands for the standard output

7 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/OSD).

– the asynchronous openFT server has not started on the partner system.

– a firewall in the partner system is blocking connections.

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:

– Were the asynchronous openFT server started?

– Does the local address match the default settings (*ftmodo -openft=@s*) or has it been changed?

– 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 */etc/hosts* 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 or *kill* on the openFT process while log records are being written.
2. File system full while writing the logging file.

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.

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*, see “Instance directory” on page 26. 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 or *kill* of openFT process with *sysfsa.** open
3. File system full on ISAM access

In cases 2 and 3, 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 by from the backup file using *ftimpe*.

– Try to repair the ISAM index file with *dcheck* (the example is valid for the standard instance):

  ```
  /opt/openFT/bin/ftbin/dcheck –b /var/openFT/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:

    ```
    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
    ```

    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

– 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.

.. 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.

Performance note

If you use the TNS during operation with CMX (*ftmodo -tns=y*), you should set the RFC1006 protocol for TNS entries in Unix systems, since the RFC1006 protocol is far more efficient than communicating via LANINET. In BS2000 systems, you should work without BCMAP entries. If you nevertheless need BCMAP entries then the following applies: If the PTSEL-I entry exists, RFC1006 is used.

In the case of operation with CMX, the RFC1006 protrocol is always used.

7.1 Actions in the event of an error

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 computer;
- the version number of the file transfer product in the remote computer, and the operating system of the remote partner computer;
- diagnostic information (which is created with the openFT command *ftshwd*).
- 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 FT profile a printout of the profile (*ftshwp␣profilename␣-l*) and a printout of the admission sets (*ftshwa␣@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)

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.

8 Diagnosis

This chapter describes how you can create and evaluate trace files. Further diagnostic information can be obtained with the help of the command
“ftshwd - Display diagnostic information” on page 291.

At the end of this chapter you will find code tables with which you can diagnose code conversion errors.

8.1 Trace files

You can switch trace mode on or off for the purposes of error diagnosis.

8.1.1 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 page 378.

When trace mode is switched on, diagnostic data is written to trace files which are located in subdirectory *traces* of the respective openFT instance, see “Instance directory” on page 26. In the case of the standard instance the path name is */var/openFT/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*.

8.1.2 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.

Files which have the suffix *.fttf* are prepared directly and are display in the openFT editor when double clicking on such a file in the openFT Explorer.

File with the suffix *.fttf* are protocol trace files. Their names begin with *Y* or *S*. Files with the suffix *.PPE* are interface trace files.

The names of the trace files have the following format:

– Y*oddhhmm*.S*ssccc*.P*ppppp*.fttf
Protocol trace files for synchronous outbound requests.

– S*oddhhmm*.S*ssccc*.I000.fttf
Protocol trace files for the control process.

– S*oddhhmm*.S*ssccc*.I*iii*.fttf
Protocol trace files for the server processes that handle asynchronous outbound requests and inbound requests.

– *process*-*pid*-*thid*-*time*.PPE
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

oddhhmm.S*ssccc*
: 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.L*iii*, e.g.:
S8101010.S33222.I001.fttf (first trace file)
S8101010.S33222.I001.L001.fttf (continuation file)

8.1.3 Evaluating trace files with fttrace

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 *-sl* is not specified.

h (high)
Passwords, user IDs, transfer admissions, account numbers, follow-up processing commands and file names are overwritten with XXX.

If the files are output in dump format (*-d*) then, irrespective of the value specified in *-sl*, the lowest security level (*-sl=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.

8.2 Code tables

8.2.1 Code table EBCDIC.DF.04

upper half byte

lower half byte	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0					SP	&	-	ø	Ø	°	µ	¢	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	î	Ã	Î	f	o	w	¶	F	O	W	6
7					å	ï	Å	Ï	g	p	x	1/4	G	P	X	7
8					ç	ì	Ç	Ì	h	q	y	1/2	H	Q	Y	8
9					ñ	ß	Ñ	¨	i	r	z	3/4	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	Û	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)

8.2.2 Code table ISO 8859-1

upper half byte

lower half byte	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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	Õ	å	õ
6			&	6	F	V	f	v			¦	¶	Æ	Ö	æ	ö
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	Ë	Ú	ê	ú
B			+	;	K	[	k	{			«	»	Ê	Û	ë	û
C			,	<	L	\	l	|			¬	1/4	Ì	Ü	ì	ü
D			-	=	M	]	m	}			SHY	1/2	Í	Ý	í	ý
E			.	>	N	^	n	~			®	3/4	Î	Þ	î	þ
F			/	?	O	_	o				¯	¿	Ï	ß	ï	ÿ

Code table ISO 8859-1

9 Appendix

This chapter contains information on

- The CSV output from the administration commands
- CMX commands
- TNS entries
- openFT cluster operation
- Administration command exit codes

9.1 Structure of CSV outputs

9.1.1 Output format

The output format for all commands corresponds to the following rules:

– Each record is output in a separate line. A record contains all the information to be displayed on an object.

– The first line is a header and contains the field names of the respective columns. **Only the field names are guaranteed, not the order of fields in the record.** In other words, the order of columns is determined by the order of the field names in the header line.

– Two tables, with their own respective headers, are output sequentially for the command *ftshwe*. If one of the tables is empty, the corresponding header is also dropped.

– Individual fields within an output line are delimited by a semicolon “;”.

The following data types are differentiated in the output:

– Number

Integer

– String

– String: Since ";" is a metacharacter in the CSV output, any text that contains ";" is enclosed in double quotes ("). Double quotes within a text field are doubled in order to differentiate them from text delimiters. When imported into a program, the doubled quotes are automatically removed and the text delimiters removed. Keywords are output in uppercase with a leading asterisk (*) and are not enclosed in double quotes.

– Date

The date and time are output in the form yyyy-mm-dd hh:mm:ss. In some cases, only the short form yyyy-mm-dd is ouput, i.e. the date alone.

– Time

The time is output in the form yyyy-mm-dd hh:mm:ss or only hh:mm:ss.

9.1.2 ftshwa

The following table indicates the CSV output format of an admission set.

The **Parameter** column contains the name of the output parameter during normal output, see page 279.

Column	Type	Values and Meaning	Parameter
UserId	String	User ID, enclosed in double quotes / *STD *STD means default admission set	USER-ID
UserMaxObs	Number	0 ... 100 Maximum user level for OUTBOUND-SEND	MAX. USER LEVELS OBS
UserMaxObsStd	String	*YES / *NO *YES means same value as default admission set [1]	
UserMaxObr	Number	0 ... 100 Maximum user level for OUTBOUND-RECEIVE	MAX. USER LEVELS OBR
UserMaxObrStd	String	*YES / *NO *YES means same value as default admission set[1]	
UserMaxIbs	Number	0 ... 100 Maximum user level for INBOUND-SEND	MAX. USER LEVELS IBS
UserMaxIbsStd	String	*YES / *NO *YES means same value as default admission set [1]	
UserMaxIbr	Number	0 ... 100 Maximum user level for INBOUND-RECEIVE	MAX. USER LEVELS IBR
UserMaxIbrStd	String	*YES / *NO *YES means same value as default admission set [1]	
UserMaxIbp	Number	0 ... 100 Maximum user level for INBOUND-PROCESSING	MAX. USER LEVELS IBP
UserMaxIbpStd	String	*YES / *NO *YES means same value as default admission set [1]	
UserMaxIbf	Number	0 ... 100 Maximum user level for INBOUND-FILE-MANAGEMENT	MAX. USER LEVELS IBF
UserMaxIbfStd	String	*YES / *NO *YES means same value as default admission set [1]	
AdmMaxObs	Number	0 ... 100 Maximum level of FTAC administrator for OUTBOUND-SEND	MAX. ADM LEVELS OBS
AdmMaxObsStd	String	*YES / *NO *YES means same value as default admission set [1]	

Column	Type	Values and Meaning	Parameter
AdmMaxObr	Number	0 ... 100 Maximum level of FTAC administrator for OUTBOUND-RECEIVE	MAX. ADM LEVELS OBR
AdmMaxObrStd	String	*YES / *NO *YES means same value as default admission set [1]	
AdmMaxIbs	Number	0 ... 100 Maximum level of FTAC administrator for INBOUND-SEND	MAX. ADM LEVELS IBS
AdmMaxIbsStd	String	*YES / *NO *YES means same value as default admission set [1]	
AdmMaxIbr	Number	0 ... 100 Maximum level of FTAC administrator for INBOUND-RECEIVE	MAX. ADM LEVELS IBR
AdmMaxIbrStd	String	*YES / *NO *YES means same value as default admission set [1]	
AdmMaxIbp	Number	0 ... 100 Maximum level of FTAC administrator for INBOUND-PROCESSING	MAX. ADM LEVELS IBP
AdmMaxIbpStd	String	*YES / *NO *YES means same value as default admission set [1]	
AdmMaxIbf	Number	0 ... 100 Maximum level of FTAC administrator for INBOUND-FILE-MANAGEMENT	MAX. ADM LEVELS IBF
AdmMaxIbfStd	String	*YES / *NO *YES means same value as default admission set [1]	
Priv	String	*YES / *NO *YES means admission set of FTAC administrator	ATTR
Password	String	*NO	ATTR
AdmPriv	String	*YES / *NO *YES means admission set of the ADM administrator	ATTR

[1] Relevant only if UserId is not *STD, *NO is always output in the case of the default admission set. In the normal output, *YES corresponds to an asterisk (*) after the value

9.1.3 ftshwatp

The following table indicates the CSV output format of an ADM trap.

The **Parameter** column indicates the name of the output parameter in the long output from *ftshwatp*, see page 286.

Column	Type	Values and Meaning	Parameter
TrapId	Number	Number of the ADM trap, up to 18 digits	TRAP-ID
Source	String	Name of the partner that triggered the trap enclosed in double quotes	SOURCE
TrapTime	Date	Date and time at which the trap occurred	DATE, TIME
TrapType	String	Type of the trap	TYPE
PartnerState	String	Partner state of the partner that triggered the trap	PTN-STATE
TransId	Number	Transfer ID [1]	TRANS-ID
RqInitiator	String	User ID or location [1] enclosed in double quotes / *REM	INITIATOR
PartnerName	String	Partner name [1] enclosed in double quotes	PARTNER
FileName	String	File name [1] enclosed in double quotes	FILENAME
RqError	String	Reason code [1] enclosed in double quotes	RC
RqErrorMsg	String	Message text [1] enclosed in double quotes	ERROR-MSG

[1] of the transfer that triggered the trap

9.1.4 ftshwc

The following table indicates the CSV output format of instances that can be remote administrated.

The **Parameter** column indicates the name of the output parameter in the normal output from *ftshwc*, see page 289.

Column	Type	Values and Meaning	Parameter
Name	String	Name enclosed in double quotes	NAME
Description	String	Description enclosed in double quotes	DESCRIPTION
Type	String	*GROUP / *INSTANCE Type (group or openFT instance)	TYPE
AccessFtAdm	String	*YES / *NO / *NONE Reading and modifying FT accesses are allowed (corresponds to FT administrator rights) / not allowed / not relevant (for Type = *GROUP)	ACCESS
AccessFtacAdm	String	*YES / *NO / *NONE Reading and modifying FTAC accesses are allowed (corresponds to FTAC administrator rights) / not allowed / not relevant (for Type = *GROUP)	ACCESS
AccessFtOp	String	*YES / *NO / *NONE Reading FT accesses are allowed / not allowed / not relevant (for Type = *GROUP)	ACCESS
Mode	String	*FTADM / *LEGACY / *NONE The instance is administered using the FTADM protocol / via ftexec / not relevant (if Type = *GROUP)	MODE

Example

```
ftshwc -csv
Name;Description;Type;AccessFtAdm;AccessFtacAdm;AccessFtOp;Mode
"Hamburg";"Northern Computer Center in Hamburg
Wandsbek";*GROUP;*NONE;*NONE;*NONE;
"Hamburg/HH1";"QA Computer Center";*GROUP;*NONE;*NONE;*NONE;
"Hamburg/HH1/HHWSRV01";"Solaris 10";*INSTANCE;*YES;*YES;*YES;*FTADM
"Hamburg/HH1/HHWSRV02";"HP-11";*INSTANCE;*YES;*YES;*YES;*FTADM
"Hamburg/HH1/HHWSRV11";"Solaris 9";*INSTANCE;*YES;*NO;*YES;*LEGACY
"Hamburg/HH2";"HR department";*GROUP;*NONE;*NONE;*NONE;
"Hamburg/HH2/HHWSRV99";"Mainframe system
(BS2000/OSD)";*INSTANCE;*NO;*NO;*YES;*FTADM
```

9.1.5 ftshwe

The command *ftshwe* sequentially displays the objects contained in an FTAC export file in a format that corresponds to the output of the *ftshwa* (page 387) and *ftshwp* (page 405) commands.

9.1.6 ftshwk

The table below indicates the CSV format for the output of the properties of the RSA keys.

The **Parameter** column contains the name of the output parameter during normal output, see page 295.

Column	Type	Values and Meaning	Parameter
Reference	Number	Key reference	KEY-REF
Identification	String	Identification of the partner enclosed in double quotes / *OWN *OWN means the private key for the user's own instance	IDENTIFICATION
CreDate	Date	Date on which the key was generated	CRE-DATE
ExpDate	String	Date on which the key expires / *NONE	EXP-DATE
Expired	String	*YES / *NO Key has expired / not expired	EXP-DATE (EXPIRED)
KeyLen	Number	768 / 1024 / 2048 Key length in bits	KEY-LEN
AuthLev	Number	1 / 2 Authentication level	AUTHL

9.1.7 ftshwl

The following table indicates the CSV output format of a log record if the option *-llf* has not been specified. If the option -llf is specified then the output has a different format, see page 395.

A format template in Microsoft Excel format is present in the following file as an example of a possible evaluation procedure:

/opt/openFT/samples/ftaccnt.xlt

The **Parameter** column contains the name of the output parameter during long output, see page 306 ff.

Column	Type	Values and Meaning	Parameter
LogId	Number	Number of the log record (up to twelve digits)	LOGGING-ID
ReasonCode	String	Reason code enclosed in double quotes to prevent interpretation as a number. FTAC Reason Codes are output as hexadecimal strings	RC
LogTime	Date	Time at which the log record was written	TIME
InitUserId	String	Initiator of the request enclosed in double quotes / *REM	INITIATOR
InitTsn	String	*NONE	---
PartnerName	String	Partner name enclosed in double quotes (name or address)	PARTNER
TransDir	String	*TO / *FROM / *NSPEC Transfer direction	TRANS
RecType	String	*FT / *FTAC / *ADM Type of log record	REC-TYPE
Func	String	*TRANS-FILE / *READ-FILE-ATTR / *DEL-FILE / *CRE-FILE / *MOD-FILE-ATTR / *READ-DIR / *MOVE-FILE / *CRE-FILE-DIR / *DEL-FILE-DIR / *LOGIN / *MOD-FILE-DIR / *REM-ADMIN / *REM-ADMIN-ROUT FT function	FUNCTION
UserAdmisId	String	User ID to which the requests in the local system relate, enclosed in double quotes	USER-ADM
FileName	String	Local file name enclosed in double quotes	FILENAME
Priv	String	*YES / *NO / *NONE Profile is privileged / not privileged / not relevant because no profile was used or no FTAC log record is present	PRIV
ProfName	String	Name of the FTAC profile enclosed in double quotes / *NONE	PROFILE

Column	Type	Values and Meaning	Parameter
ResultProcess	String	*STARTED / *NOT-STARTED / *NONE Status of follow-up processing	PCMD
StartTime	Date	Start time of transfer	STARTTIME
TransId	Number	Number of transfer request	TRANS-ID
Write	String	*REPL / *EXT / *NEW / *NONE Write rules	WRITE
StoreTime	Date	Acceptance time of request – If initiated in the local system: time the request was issued – If initiated in the remote system: time of entry in the request queueh	REQUESTED STORETIME
ByteNum	Number	Number of bytes transferred	TRANSFER
DiagInf	String	Diagnostic information / *NONE	---
ErrInfo	String	Additional information on the error message, enclosed in double quotes / *NONE	ERRINFO
Protection	String	*SAME / *STD Protection attributes are transferred / not transferred	PROTECTION ---
ChangeDate	String	*SAME / *STD Take over modification date of send file for receive file / do not take over modification date	CHG-DATE
SecEncr	String	*YES / *NO Encryption of request description activated / deactivated	SEC-OPTS
SecDichk	String	*YES / *NO Data integrity check of request description activated / deactivated	SEC-OPTS
SecDencr	String	*YES / *NO Encryption of transferred file content activated / deactivated	SEC-OPTS
SecDdichk	String	*YES / *NO Data integrity check of transferred file content activated / deactivated	SEC-OPTS
SecLauth	String	*YES / *NO Authentication of the local system in the remote system activated / deactivated	SEC-OPTS
SecRauth	String	*YES / *NO Authentication of the remote system in the local system activated / deactivated	SEC-OPTS
RsaKeyLen	Number	768 / 1024 / 2048 / empty Length of the RSA key used for the encryptio in bit or empty if SecEncr does not have the value *YES	SEC-OPTS

Column	Type	Values and Meaning	Parameter
SymEncrAlg	String	*DES / *AES-128 / *AES-256 / empty The encryption algorithm used or empty if SecEncr does not have the value *YES	SEC-OPTS
CcsName	String	Name of the character set enclosed in double quotes / empty	CCS-NAME
AdminId	String	Administrator ID on the remote administration server, enclosed in double quotes / empty	ADMIN-ID
Routing	String	Routing information enclosed in double quotes / empty	ROUTING
AdmCmd	String	Administration kommand enclosed in double quotes / empty	ADM-CMD
As3Type	String	empty (internal function)	---
As3MsgTid	String	empty (internal function)	---
As3RcpStat	String	empty (internal function)	---
AuthLev	Number	1 / 2 / empty Authentication level	SEC-OPTS
GlobReqId	Number	Global request identification (requests issued remotely) / empty (requests issued locally)	GLOB-ID

CSV output on ftshwl -llf

If the option *-llf* is specified then only the following columns are output:

Column	Type	Values and Meaning	Parameter
TimeStamp	Date	Creation time of the log file	---
LoggingFileName	String	Fully qualified name of the log file	(file name)

9.1.8 ftshwm

The following table shows the CSV output format for the monitoring values for openFT operation if all the monitoring values are output (*ftshwm -csv @a*).

If the *-raw* option is specified, the duration values are not output (*Duxxx*, see footnote).

The default values are marked with "x" in the **Std** column. These are output if *ftshwm -csv* is specified without *@a* and without names being specified explicitly.

For a detailed description of the monitoring values, refer to the section “Description of the monitoring values” on page 324.

The individual monitoring values (ThNetbTtl ... StTrcr) have the same names in all the output formats (normal output, long output and CSV output).

Column	Type	Values prepared	Values not prepared	Meaning	Std
CurrTime	Date	Time	Time	Current timet	x
MonOn	Date	Time	Time	Start time of measurement date recording or last change of configuration (a modification of PartnerSel/ReqSel has the same effect as a new start)	x
PartnerSel	String6	*ALL / *NONE / OPENFT / FTAM / FTP		Partner type selected	x
ReqSel	String	*ALL / ONLY-SYNC / ONLY-ASYNC / ONLY-LOCAL / ONLY-REMOTE		Request type selected	x
Data	String	FORM	RAW	Output format (perpared / not prepared)	x
ThNetbTtl	Number	Number of bytes per second	Bytes, accumulated	Throughput in net bytes	x
ThNetbSnd	Number	Number of bytes per second	Bytes, accumulated	Throughput in net bytes, send requests	x
ThNetbRcv	Number	Number of bytes per second	Bytes, accumulated	Throughput in net bytes, receive requests	x
ThNetbTxt	Number	Number of bytes per second	Bytes, accumulated	Throughput in net bytes, text files	
ThNetbBin	Number	Number of bytes per second	Bytes, accumulated	Throughput in net bytes, binary files	
ThDiskTtl	Number	Number of bytes per second	Bytes, accumulated	Throughput in disk bytes	x
ThDiskSnd	Number	Number of bytes per second	Bytes, accumulated	Throughput in disk bytes, send requests	x

Column	Type	Values prepared	Values not prepared	Meaning	Std
ThDiskRcv	Number	Number of bytes per second	Bytes, accumulated	Throughput in disk bytes, receive requests	x
ThDiskTxt	Number	Number of bytes per second	Bytes, accumulated	Throughput in disk bytes, text files	
ThDiskBin	Number	Number of bytes per second	Bytes, accumulated	Throughput in disk bytes, binary files	
ThRqto	Number	Number per second	Number, accumulated	openFT requests received	x
ThRqft	Number	Number per second	Number, accumulated	File transfer requests received	
ThRqfm	Number	Number per second	Number, accumulated	file management requests received	
ThSuct	Number	Number per second	Number, accumulated	Successfully completed openFT requests	x
ThAbrt	Number	Number per second	Number, accumulated	Aborted openFT requests	x
ThIntr	Number	Number per second	Number, accumulated	Interrupted openFT requests	x
ThUsrf	Number	Number per second	Number, accumulated	Requests from non-authorized users	x
ThFoll	Number	Number per second	Number, accumulated	Follow-up processing operations started	
ThCosu	Number	Number per second	Number, accumulated	Connections established	
ThCofl	Number	Number per second	Number, accumulated	Failed connection attempts	x
ThCobr	Number	Number per second	Number, accumulated	Disconnections as a result of connection errors	x
DuRqtlOut[1]	Number	Milliseconds	---	Maximum request duration Outbound	
DuRqtlInb[1]	Number	Milliseconds	---	Maximum request duration Inbound	
DuRqftOut[1]	Number	Milliseconds	---	Maximum request duration Outbound transfer	
DuRqftInb[1]	Number	Milliseconds	---	Maximum request duration Intbound transfer	
DuRqfmOut[1]	Number	Milliseconds	---	Maximum request duration Outbound file management	

Column	Type	Values prepared	Values not prepared	Meaning	Std
DuRqfmInb[1]	Number	Milliseconds	---	Maximum request duration Inbound file management	
DuRqesOut[1]	Number	Milliseconds	---	Maximum outbound request waiting time	
DuDnscOut[1]	Number	Milliseconds	---	Maximum time an outbound openFT request was waiting for partner checking	
DuDnscInb[1]	Number	Milliseconds	---	Maximum time an inbound openFT request was waiting for partner checking	
DuConnOut[1]	Number	Milliseconds	---	Maximum duration tim of establishment of a connection for an outbound openFT request	
DuOpenOut[1]	Number	Milliseconds	---	Maximum file open time (outbound)	
DuOpenInb[1]	Number	Milliseconds	---	Maximum file open time (inbound)	
DuClosOut[1]	Number	Milliseconds	---	Maximum file close time (outbound)	
DuClosInb[1]	Number	Milliseconds	---	Maximum file close time (inbound)	
DuUsrcOut[1]	Number	Milliseconds	---	Maximum user check time (outbound)	
DuUsrcInb[1]	Number	Milliseconds	---	Maximum user check time (ínbound)	
StRqas	Number (100)[2]	Average value	Current number	Number of synchronous requests in the ACTIVE state	x
StRqaa	Number (100)[2]	Average value	Current number	Number of asynchronous requests in the ACTIVE state	x
StRqwt	Number (100)[2]	Average value	Current number	Number of requests in the WAIT state	x
StRqhd	Number (100)[2]	Average value	Current number	Number of requests in the HOLD state	x
StRqsp	Number (100)[2]	Average value	Current number	Number of requests in the SUSPEND state	x
StRqlk	Number (100)[2]	Average value	Current number	Number of requests in the LOCKED state	x
StRqfi	Number (100)[2]	Average value	Current number	Number of requests in the FINISHED state	

Column	Type	Values prepared	Values not prepared	Meaning	Std
StCLim	Number	Value currently set		Maximum number of connections established for asynchronous requests.	x
StCAct	Percent	Share of StCLim in %	Current number	Number of occupied connections for asynchronous requests	x
StRqLim	Number	Value currently set		Maximum number of asynchronous requests in request management	x
StRqAct	Percent	Share of StRqLim in %	Current number	Entries occupied in request management	x
StOftr	BOOL	1 / 0		openFT Protocol activated / deactivated	x
StFtmr	BOOL	1 / 0		FTAM Protocol activated / deactivated	x
StFtpr	BOOL	1 / 0		FTP Protocol activated / deactivated	x
StTrcr	BOOL	1 / 0		Trace activated / deactivated	

[1] is not output with option *-raw*

[2] number is the measured value multiplied by 100 (e.g. output 225 corresponds to value 2.25)

Examples

```
ftshwm -ty -csv @a
```

```
CurrTime;MonOn;PartnerSel;ReqSel;Data;ThNetbTtl;ThNetbSnd;ThNetbRcv;ThNetbTxt
;ThNetbBin;ThDiskTtl;ThDiskSnd;ThDiskRcv;ThDiskTxt;ThDiskBin;ThRqto;ThRqft;Th
Rqfm;ThSuct;ThAbrt;ThIntr;ThUsrf;ThFoll;ThCosu;ThCofl;ThCobr;DuRqtlOut;DuRqtl
Inb;DuRqftOut;DuRqftInb;DuRqfmOut;DuRqfmInb;DuRqesOut;DuDnscOut;DuDnscInb;DuC
onnOut;DuOpenOut;DuOpenInb;DuClosOut;DuClosInb;DuUsrcOut;DuUsrcInb;StRqas;StR
qaa;StRqwt;StRqhd;StRqsp;StRqlk;StRqfi;StCLim;StCAct;StRqLim;StRqAct;StOftr;S
tFtmr;StFtpr;StTrcr
*TIME;*TIME;*STRING;*STRING;*STRING;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*
INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*IN
T;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT(100);
*INT(100);*INT(100);*INT(100);*INT(100);*INT(100);*INT(100);*INT;*PERCENT;*IN
T;*PERCENT;*BOOL;*BOOL;*BOOL;*BOOL
```

```
ftshwm -csv ThNetbTtl ThDiskTtl
```

```
CurrTime;MonOn;PartnerSel;ReqSel;Data;ThNetbTtl;ThDiskTtl
2008-02-28 15:40:01;2008-02-28 15:36:12;OPENFT,FTAM;ONLY-ASYNC,ONLY-
REMOTE;FORM;2681262;524064
```

9.1.9 ftshwo

The following table indicates the CSV output format of the operating parameters

The **Parameter** column contains the name of the output parameter during normal output, see page 331 ff. Some parameters have fixed values because they are supported only for reasons of compatibility or have been replaced by other parameters.

Column	Type	Values and Meaning	Parameter
PartnerLim	Number	0	---
ReqLim	Number	Maximum number of requests	RQ-LIM
TaskLim	Number	Maximum number of processes	PROC-LIM
ConnLim	Number	Maximum number of connections	CONN-LIM
ReqWaitLev	Number	1	---
TransportUnitSize	Number	Maximum length of a transport unit	TU-SIZE
PartnerCheck	String	*STD / *TRANSP-ADDR Partner check	PTN-CHK
SecLev	Number	0... 100 / *B-P-ATTR Default value for the security level of partners	SEC-LEV
TraceOpenft	String	*STD / *OFF Trace function for openFT partner activated / deactivated	FUNCT, line TRACE PARTNER-SELECTION
TraceOut	String	*FILE / empty Trace function activated / deactivated	FUNCT, line TRACE SWITCH---
TraceSession	String	*OFF	---
TraceFtam	String	*STD / *OFF Trace function for FTAM partner activated / deactivated	FUNCT, line TRACE PARTNER-SELECTION
LogTransFile	String	*ON / *OFF FT logging activated / deactivated	FT-LOG
MaxInboundReq	Number	Maximum number of requests	(same as RQ-LIM)
MaxReqLifetime	String	Maximum lifetime of requests in the request queue / *UNLIMITED	MAX-RQ-LIFE
SnmpTrapsSubsystemState	String	*ON / *OFF SNMP traps on subsystem status change activated / deactivated	TRAP, line SNMP SS-STATE
SnmpTrapsFtState	String	*ON / *OFF SNMP traps on asynchronous server status change activated / deactivated	TRAP, line SNMP FT-STATE

Column	Type	Values and Meaning	Parameter
SnmpTrapsPartnerState	String	*ON / *OFF SNMP traps on partner status change activated / deactivated	TRAP, line SNMP PART-STATE
SnmpTrapsPartnerUnreach	String	*ON / *OFF SNMP traps on unreachable partner systems activated / deactivated	TRAP, line SNMP PART-UNREA
SnmpTrapsReqQueueState	String	*ON / *OFF SNMP traps on request management status change activated / deactivated	TRAP, line SNMP RQ-STATE
SnmpTrapsTransSucc	String	*ON / *OFF SNMP traps on successfully terminated requests activated / deactivated	TRAP, line SNMP TRANS-SUCC
SnmpTrapsTransFail	String	*ON / *OFF SNMP traps on failed requests activated / deactivated	TRAP, line SNMP TRANS-FAIL
ConsoleTraps	String	*ON / *OFF Console traps (for at least one criterion) activated / deactivated.	TRAP, line CONS
TeleService	String	empty	
HostName	String	Host name of the local computer / *NONE	HOST-NAME
Identification	String	Instance identification enclosed in double quotes	IDENTIFICATION
UseTns	String	*YES / *NO Use / do not use TNS in operation with CMX	USE TNS
ConsTrapsSubsystemState	String	*ON / *OFF Console traps on subsystem status change activated / deactivated	TRAP, line CONS SS-STATE
ConsTrapsFtState	String	*ON / *OFF Console traps on asynchronous server status change activated / deactivated	TRAP, line CONS FT-STATE
ConsTrapsPartnerState	String	*ON / *OFF Console traps on partner status change activated / deactivated	TRAP, line CONS PART-STATE
ConsTrapsPartnerUnreach	String	*ON / *OFF Console traps on unreachable partner systems activated / deactivated	TRAP, line CONS PART-UNREA
ConsTrapsReqQueueState	String	*ON / *OFF Console traps on request management status change activated / deactivated	TRAP, line CONS RQ-STATE

Column	Type	Values and Meaning	Parameter
ConsTrapsTransSucc	String	*ON / *OFF Console traps on successfully terminated requests activated / deactivated	TRAP, line CONS TRANS-SUCC
ConsTrapsTransFail	String	*ON / *OFF Console traps on failed requests activated / deactivated	TRAP, line CONS TRANS-FAIL
FtLog	String	*ALL / *FAIL / *NONE Scope of FT logging	FT-LOG
FtacLog	String	*ALL / *FAIL / *NONE Scope of FTAC logging	FTAC-LOG
Trace	String	*ON / *OFF Trace function activated / deactivated	FUNCT, line TRACE SWITCH
TraceSelp	String	*ALL / OPENFT / FTP / FTAM / ADM / empty [1] Trace selection based on partner type	FUNCT, line TRACE PARTNER-SELECTION
TraceSelr	String	*ALL / ONLY-SYNC / ONLY-ASYNC / ONLY-LOCAL / ONLY-REMOTE [1] Trace selection based on request type	FUNCT, line TRACE REQUEST-SELECTION
TraceOpt	String	*NO-BULK-DATA / *NONE Minimum trace / no trace options	FUNCT, line TRACE OPTIONS
KeyLen	Number	768 / 1024 / 2048 RSA key length in bit	KEY-LEN
CcsName	String	Character set enclosed in double quotes	CCS-NAME
AppEntTitle	String	*YES / *NO In the case of FTAM, "nil-Application Entity Title" is sent / not sent	---
StatName	String	Name of the local openFT application$FJAM	LOCAL-SYSTEM-NAME
SysName	String	Name of the local system / empty	LOCAL-SYSTEM-NAME
FtStarted	String	*YES / *NO Asynchronous openFT server started / not started	STARTED
openftAppl	String	*STD / port number Port number of the local openFT server	OPENFT-APPL
ftamAppl	String	*STD / port number Port number of the local FTAM server	FTAM-APPL
FtpPort	Number	Port number Port number of the local FTP server	FTP-PORT
ftpDPort	Number	Value / empty (internal function)	---
ftstdPort	String	*STD / port number Default port for dynamic partners	---

Column	Type	Values and Meaning	Parameter
DynPartner	String	*ON / *OFF Dynamic partner entries activated / deactivated	DYN-PART
ConTimeout	Number	Value (internal function)	---
ChkpTime	Number	Value (internal function)	---
Monitoring	String	*ON / *OFF Monitoring data activated / deactivated	FUNCT, line MONITOR SWITCH
MonSelp	String	*ALL / OPENFT / FTP / FTAM / empty [1] Selection based on type of partner system	FUNCT, line MONITOR PARTNER-SELECTION
MonSelr	String	*ALL / ONLY-SYNC / ONLY-ASYNC / ONLY-LOCAL / ONLY-REMOTE [1] Selection based on type of request	FUNCT, line MONITOR REQUEST-SELECTION
AdmTrapServer	String	Name of the ADM-TRAP server / *NONE	ADM-TRAP-SERVER
AdmTrapsFtState	String	*ON / *OFF ADM traps on asynchronous server status change activated / deactivated	TRAP, line ADM FT-STATE
AdmTrapsPartnerState	String	*ON / *OFF ADM traps on partner status change activated / deactivated	TRAP, line ADM PART-STATE
AdmTrapsPartnerUnreach	String	*ON / *OFF ADM traps on unreachable partner systems activated / deactivated	TRAP, line ADM PART-UNREA
AdmTrapsReqQueueState	String	*ON / *OFF ADM traps on request management status change activated / deactivated	TRAP, line ADM RQ-STATE
AdmTrapsTransSucc	String	*ON / *OFF ADM traps on successfully terminated requests activated / deactivated	TRAP, line ADM TRANS-SUCC
AdmTrapsTransFail	String	*ON / *OFF ADM traps on failed requests activated / deactivated	TRAP, line ADM TRANS-FAIL
AdminConnLim	String	Maximum number of administration connections	ADM-CLIM
AdmPort	String	Port number / *NONE Port number for remote administration	ADM-PORT
OpenftApplState	String	*ACTIVE / *INACT / *DISABLED / *NAVAIL Status of the openFT server	OPENFT-APPL, 2nd line
FtamApplState	String	*ACTIVE / *INACT / *DISABLED / *NAVAIL Status of the FTAM server	FTAM-APPL, 2nd line

Column	Type	Values and Meaning	Parameter
FtpState	String	*ACTIVE / *INACT / *DISABLED / *NAVAIL Status of the FTP server	FTP-PORT, 2nd line
AdmState	String	*ACTIVE / *INACT / *DISABLED Status for inbound remote administration, on ADM trap server also status for receiving ADM traps	ADM-PORT, 2nd line
AdminLog	String	*ALL / *FAIL / *MODIFY / *NONE Scope of ADM logging	ADM-LOG
CentralAdminServer	String	*YES / *NO Local computer is remote administration server / not remote administration server	ADM-CS
ActiveAppl	String	*ALL / *NONE / OPENFT / FTAM / FTP / ADM[1] active servers	see 2nd line of OPENFT-APPL, FTAM-APPL, FTP-PORT, ADM-PORT
UseCmx	String	*YES / *NO Operation with CMX / without CMX	USE CMX
TraceOptLowerLayers	String	*DETAIL / *STD / *OFF Trace scope for lower protocol layers	OPTIONS-LL
EncMandIn	String	*YES / *NO Inbound encryption activated / deactivated	ENC-MAND (IN)
EncMandOut	String	*YES / *NO Outbound encryption activated / deactivated	ENC-MAND (OUT)
DelLog	String	*ON / *OFF Automatic deletion of log records activated / deactivated	DEL-LOG
DelLogRetpd	Number	Minimum age, in days, of the log records to be deleted. 0 means current day.	RETPD
DelLogRepeat	String	*MONTHLY / *WEEKLY / *DAILY Repeat interval for deletion of log records.	DEL-LOG ON
DelLogDay	Number	1..31 / 1..7 / 0 Day on which deletion is to be repeated. In the case of DelLogRepeat = *MONTHLY then this is the day of the month, if DelLogRepeat = *WEEKLY then it is the day of the week (1 = Monday), if DelLogRepeat = *DAILY then 0 is output	DEL-LOG ON
DelLogTime	Time	Time of deletion	DEL-LOG AT

[1] Combinations of multiple values are also possible (not with *ALL or *NONE)

9.1.10 ftshwp

The following table indicates the CSV output format of an admission profile.

The **Parameter** column contains the name of the output parameter during long output, see also page 339f and page 340f.

Column	Type	Values and Meaning	Parameter
ProfName	String	Name of the profile enclosed in double quotes	(Profile name)
Priv	String	*YES / *NO Profile is privileged / not privileged	PRIVILEGED
TransAdm	String	*SECRET / *NSPEC Transfer admission has been assigned / not assigned	TRANS-ADM NOT-SPECIFIED
Duplicated	String	*YES / *NO *YES means: profile is locked due to attempt to assign the transfer admission twice	TRANS-ADM DUPLICATED
LockedByImport	String	*YES / *NO *YES means: profile is locked because it was imported	TRANS-ADM LOCKED (by_import)
LockedByAdm	String	*YES / *NO *YES means: profile locked by FTAC administrator	TRANS-ADM LOCKED (by_adm)
LockedByUser	String	*YES / *NO *YES means: profile locked by user	TRANS-ADM LOCKED (by_user)
Expired	String	*YES / *NO *YES means: profile locked because period expired	TRANS-ADM EXPIRED
ExpDate	String	Expiration date in short format yyyy-mm-dd / *NRES (no expiration date)	EXP-DATE
Usage	String	*PUBLIC / *PRIVATE / *NSPEC Usage	USAGE
IgnObs	String	*YES / *NO Ignore / do not ignore predefined value for Outbound Send	IGN-MAX-LEVELS OBS
IgnObr	String	*YES / *NO Ignore / do not ignore predefined value for Outbound Receive	IGN-MAX-LEVELS OBR
IgnIbs	String	*YES / *NO Ignore / do not ignore predefined value for Inbound Send	IGN-MAX-LEVELS IBS

Column	Type	Values and Meaning	Parameter
IgnIbr	String	*YES / *NO Ignore / do not ignore predefined value for Intbound Receive	IGN-MAX-LEVELS IBR
IgnIbp	String	*YES / *NO Ignore / do not ignore predefined value for Inbound Processing	IGN-MAX-LEVELS IBP
IgnIbf	String	*YES / *NO Ignore / do not ignore predefined value for Inbound File Management	IGN-MAX-LEVELS IBF
Initiator	String	*LOC / *REM / *NRES Initiator: only local / only remote / unrestricted	INITIATOR
TransDir	String	*FROM / *TO / *NRES Permitted transfer direction: from partner / to partner / unrestricted	TRANS-DIR
MaxPartLev	Number	0... 100 / *NRES Maximum security level / security level unrestrictedt	MAX-PART-LEV
Partners	String	One or more FT partners, delimited by commas and enclosed in double quotes / *NRES (no restriction)	PARTNER
FileName	String	File name or file name prefix enclosed in double quotes / *NRES Restricts access to this file or files with this prefix. *NRES means there is no restriction	FILE-NAME
Library	String	*NRES not relevant on Unix systems	LIBRARY
FileNamePrefix	String	*YES / *NO The file name in FileName is a prefix / is not a prefix	FILE-NAME = (PREFIX=..)
ElemName	String	*NRES	---
ElemPrefix	String	*NO	---
ElemVersion	String	*NRES	---
ElemType	String	*NRES	---
FilePass	String	*NRES	---
Write	String	*NEW / *EXT / *REPL / *NRES Write rules	WRITE
UserAdmId	String	User ID enclosed in double quotes	USER-ADM (user-id,...)

Column	Type	Values and Meaning	Parameter
UserAdmAcc	String	Account number enclosed in double quotes / *NRES	USER-ADM (..,account,...)
UserAdmPass	String	*OWN / *YES / *NSPEC / *NONE Password is taken over / was specified / was not specified / is not required	USER-ADM (...,...,password)
ProcAdmId	String	*NRES	---
ProcAdmAcc	String	*NRES	---
ProcAdmPass	String	*NRES	---
SuccProc	String	Follow-up processing on success, enclosed in double quotes / *NONE / *NRES / *EXPANSION	SUCC-PROC
SuccPrefix	String	Folow-up processing prefix on success, enclosed in double quotes / *NONE	SUCC-PREFIX
SuccSuffix	String	Follow-up processing suffix on success, enclosed in double quotes / *NONE	SUCC-SUFFIX
FailProc	String	Follow-up processing on error, enclosed in double quotes / *NONE / *NRES / *EXPANSION	FAIL-PROC
FailPrefix	String	Follow-up processing prefix on error, enclosed in double quotes / *NONE	FAIL-PREFIX
FailSuffix	String	Follow-up processing suffix on error, enclosed in double quotes / *NONE	FAIL-SUFFIX
TransFile	String	*ALLOWED / *NOT-ALLOWED Transfer and delete files permitted / not permitted	FT-FUNCTION = (TRANSFER-FILE)
ModFileAttr	String	*ALLOWED / *NOT-ALLOWED Modify file attributes permitted / not permitted	FT-FUNCTION = (MODIFY-FILE-ATTRIBUTES)
ReadDir	String	*ALLOWED / *NOT-ALLOWED View directories permitted / not permitted	FT-FUNCTION = (READ-DIRECTORY)
FileProc	String	*ALLOWED / *NOT-ALLOWED Preprocessing/postprocessing permitted / not permitted	FT-FUNCTION = (FILE-PROCESSING)
AccAdm	String	*ALLOWED / *NOT-ALLOWED Access to remote administration server permitted / not permitted	FT-FUNCTION = (ACCESS-TO-ADMINISTRATION)
RemAdm	String	*ALLOWED / *NOT-ALLOWED Remote administration via remote administration server permitted / not permitted	FT-FUNCTION = (REMOTE-ADMINISTRATION)
Text	String	Text enclosed in double quotes / *NONE	TEXT

Column	Type	Values and Meaning	Parameter
DataEnc	String	*YES / *NO / *NRES Data encryption is mandatory / prohibited / neither mandatory nor prohibited	DATA-ENC
ModDate	Date	Time of last modification	LAST-MODIF
AdmTrapLog	String	*ALLOWED / *NOT-ALLOWED Reception of ADM traps permitted / not permitted	FT-FUNCTION = (ADM-TRAP-LOG)

9.1.11 ftshwptn

The following table indicates the CSV output format of a partner in the partner list.

The **Parameter** column contains the name of the output parameter during long output, see page 345.

Column	Type	Values and Meaning	Parameter
PartnerName	String	Partner name enclosed in double quotes	NAME
Sta	String	*ACT / *DEACT / *NOCON / *LUNK / *RUNK / *ADEAC / *AINAC / *LAUTH / *RAUTH / *NOKEY / *DIERR / *IDREJ Partner status	STATE
SecLev	String	*STD / *B-P-ATTR / 1...100 Global security level / attribute-specific security level / fixed security level	SECLEV
Trace	String	*FTOPT / *STD / *ON / *OFF Trace setting	TRACE
Loc	Number	Number of locally issued file transfer requests to this partner	LOC
Rem	Number	Number of file transfer requests issued by this partner	REM
Processor	String	Processor name enclosed in double quotes / empty	ADDRESS
Entity	String	Entity name enclosed in double quotes / empty	ADDRESS
NetworkAddr	String	Partner address (network address without port number/selectors) enclosed in double quotes	ADDRESS
Port	Number	Port number	ADDRESS (port number)
PartnerCheck	String	*FTOPT / *STD / *TRANSP-ADDR / *AUTH / *AUTHM / *NOKEY Sender verification	P-CHK
TransportSel	String	Transport selector enclosed in double quotes / empty	ADDRESS (transport selector)
LastAccessDate	Date	Time of last access in short format yyyy-mm-dd	---
SessionSel	String	Session selector enclosed in double quotes / empty	ADDRESS (session selector)
PresentationSel	String	Presentation selector enclosed in double quotes / empty	ADDRESS (presentation selector)
Identification	String	Identification enclosed in double quotes / empty	IDENTIFICATION

Column	Type	Values and Meaning	Parameter
SessRout	String	Routing information enclosed in double quotes / *ID / empty *ID means routing information same as identification	ROUTING
PartnerAddr	String	Partner address (including port number und selectors) enclosed in double quotes	ADDRESS
Check	String	*FTOPT / *STD / *TRANSP-ADDR Partner check	P-CHK
AuthMand	String	*YES / *NO Authentication is mandatory / not mandatory	P-CHK
Priority	String	*LOW / *NORM / *HIGH Priority	PRI
AS3	String	*NO (internal function)	---
AuthLev	Number	1 / 2 / empty Authentication level	P-CHK
InboundSta	String	*ACT / *DEACT Inbound function activated / deactivated	INBND
RequProc	String	*STD / *SERIAL The processing mode for asynchronous outbound requests is parallel / is serial	REQU-P

9.1.12 ftshwr

The following table indicates the CSV output format of a request.

Short output is also possible with *ftshwr*, see page 414.

The **Parameter** column contains the name of the output parameter during long output, see page 352.

Column	Type	Values and Meaning	Parameter
TransId	Number	Request ID	TRANSFER-ID
Initiator	String	*LOC / *REM Initiator is local / remote	INITIATOR
State	String	*LOCK / *WAIT / *HOLD / *FIN / *ACT / *CANC / *SUSP Request status	STATE
PartnerName	String	Name or address of the partner enclosed in double quotes	PARTNER
PartnerState	String	*ACT / *INACT / *NOCON / *INSTERR Partner status	PARTNER-STATE
TransDir	String	*TO / *FROM Transfer direction	TRANS
ByteNum	Number	Number of bytes transferred / empty	BYTECNT
LocFileName	String	File name in the local system enclosed in double quotes	LOC: FILE
LocElemName	String	empty	---
LocElemType	String	empty	---
LocElemVersion	String	empty	---
Prio	String	*NORM / *LOW Priority of the request	PRIO
Compress	String	*NONE / *BYTE / *ZIP Compressed transfer	COMPRESS
DataEnc	String	*YES / *NO User data is transferred encrypted / unencrypted	ENCRYPT
DiCheck	String	*YES / *NO Data integrity is checked / is not checked	DICHECK
Write	String	*REPL / *EXT / *NEW Write rules	WRITE
StartTime	String	Time at which the request is started (format yy-mm-dd hh:mm:ss) / *SOON (request is started as soon as possible)	START

Column	Type	Values and Meaning	Parameter
CancelTime	String	Time at which the request is deleted from the request queue (format yy-mm-dd hh:mm:ss) / *NO (no delete time)	CANCEL
Owner	String	Local user ID enclosed in double quotes	OWNER
DataType	String	*CHAR / *BIN / *USER File type	DATA
Transp	String	*YES / *NO Transfer transparent / not transparent	TRANSP
LocTransAdmId	String	User ID for accessing the local system, enclosed in double quotes / *NONE	LOC: TRANS-ADM (USER)
LocTransAdmAcc	String	empty	---
LocProfile	String	empty	---
LocProcAdmId	String	empty	---
LocProcAdmAcc	String	empty	---
LocSuccProc	String	Local follow-up processing on success, enclosed in double quotes / *NONE / empty	LOC: SUCC-PROC
LocFailProc	String	Local follow-up processing on error, enclosed in double quotes / *NONE / empty	LOC: FAIL-PROC
LocListing	String	empty	---
LocMonjv	String	empty	---
LocCcsn	String	Name of the character set in the local system enclosed in double quotes / *STD	LOC: CCSN
RemFileName	String	File name in the remote system enclosed in double quotes / *NSPEC / *NONE / empty	REM: FILE
RemElemName	String	empty	---
RemElemType	String	empty	---
RemElemVersion	String	empty	---
RemTransAdmId	String	User ID in the remote system enclosed in double quotes / *NONE	REM: TRANS-ADM=(user-id,...)
RemTransAdmAcc	String	Account number in the remote system enclosed in double quotes / empty	REM: TRANS-ADM=(...,account)
RemTransAdmAccount [1]	String	Account number in the remote system enclosed in double quotes / empty	REM: TRANS-ADM=(...,account)

Column	Type	Values and Meaning	Parameter
RemProfile	String	*YES / *NONE *YES means access via FTAC admission profile	REM: TRANS-ADM=REMOTE-PROFILE
RemProcAdmId	String	empty	---
RemProcAdmAcc	String	empty	---
RemSuccProc	String	Remote follow-up processing on success, enclosed in double quotes / *NONE / empty	REM: SUCC-PROC
RemFailProc	String	Remote follow-up processing on error, enclosed in double quotes / *NONE / empty	REM: FAIL-PROC
RemCcsn	String	Name of the character set used in the remote system, enclosed in double quotes / *STD	REM: CCSN
FileSize	Number	Size of the file in bytes / empty	FILESIZE
RecSize	Number	Maximum record size in bytes / empty	RECSIZE
RecFormat	String	*STD / *VARIABLE / *FIX / *UNDEFINED Record format	RECFORM
StoreTime	Date	Time at which the request was entered in the request queue	STORE
ExpEndTime	Date	empty	---
TranspMode	String	*YES / *NO Transfer transparent / not transparent	TRANSP
DataEncrypt	String	*YES / *NO User data transferred encrypted / unencrypted	ENCRYPT
TabExp	String	*AUTO / *YES / *NO Tabulator expansion	TABEXP
Mail	String	*ALL / *FAIL / *NO Result messages	LOC: MAIL
DiagCode	String	Diagnostic information / empty	DIAGCODE
FileAvail	String	*IMMEDIATE / *DEFERRED / *NSPEC Availability (for FTAM only)	AVAILABILITY
StorageAccount	String	Account number (for FTAM only) / empty	STOR-ACCOUNT
AccessRights	String	FTAM access rights / empty Possible values are @r, @w or combinations of r, i, p, x, e, a, c, d	ACCESS-RIGHTS

Column	Type	Values and Meaning	Parameter
LegalQualif	String	Legal qualification (for FTAM only) / empty	LEGAL-QUAL
PartnerPrio	String	*LOW / *NORM / *HIGH Partner priority	PARTNER-PRIO
TargetFileForm	String	*STD / *BLOCK / *SEQ File format in the target system	TARGFORM
TargetRecForm	String	*STD / *UNDEFINED Record format in the target system	TRECFRM
Protection	String	*STD / *SAME Transfer of protection attributes	PROTECT
GlobReqId	Number	Global request identification For locally issued requests, same as request ID; for globally issued requests, same as the request ID in the initiating system	TRANSFER-ID or GLOB-ID

[1] RemTransAdmAcc and RemTransAdmAccount have the same meaning and the same content. For reasons of compatibility, both parameters are present in the CSV output.

Short output from ftshwr in CSV format

ftshwr -s -csv outputs a table with two rows indicating the number of requests that have the corresponding status, see also page 352.

Column	Type	Values
Act	Number	Number of requests with the status ACTIVE
Wait	Number	Number of requests with the status WAIT
Lock	Number	Number of requests with the status LOCK
Susp	Number	Number of requests with the status SUSPEND
Hold	Number	Number of requests with the status HOLD
Fin	Number	Number of requests with the status FINISHED
Total	Number	Total number of requests

Example

```
ftshwr -s -csv

Act;Wait;Lock;Susp;Hold;Fin;Total
0;1;0;0;2;0;3
```

9.2 Important CMX commands

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“.

tnsxcom - Create the TS directory

With the *tnsxcom* 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):

tnsxcom [**-l -s -S -u -i**] [file]

The options have the following meanings:

-l LOAD mode
tnsxcom takes the entries our of the file *file* one at a time and fills the (previously empty) TS directory with the syntactically correct entries.

-s CHECK mode
tnsxcom 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 *tnsxcom* updates the TS directory in a second run.

-u UPDATE mode
tnsxcom takes the entries our of 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 INTERAKTIVE mode
tnsxcom 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:

```
tnsxcom –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: `tnsxcom –u upd.dir`

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 the file *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_RESTRICTED 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
   ```

9.3 Entering transport system applications in the TNS

As of openFT V10, it is no longer 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 the section “TNS entries created automatically” on page 420.)

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:

Call the script */opt/openFT/bin/ftbin/ftgentns*.

Creating TNS entries manually

The entries in the TNS can be made with the aid of the TNS compilers *tnsxcom*. To do this, enter the TS applications in a file, and then translate this file with the aid of the TNS compilers *tnsxcom* (see the section “tnsxcom - Create the TS directory” on page 416).

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.

It can also be useful to enter the remote TS applications of the partner systems which are to issue requests to the local system. In openFT partner version 8.1 and later, 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 TCOP/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 on page 423.

TNS entries for cluster configurations

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 an example of this You will find two examples of this on page 428 and page 433.

9.3.1 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 (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'
```

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.

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.

The same also applies if a version of openFT < V10.0 was installed, as TNS entries are not deleted on uninstallation.

9.3.2 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.

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.

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
```

9.3.3 Definition of a remote TS application for openFT

In openFT partners with version 8.1 and later, 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 file translated using the TNS compiler *tnsxcom* or the graphical interface (if present).

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.

9.3.3.1 Sample entries for openFT partners

The examples listed below assume that the corresponding transport system is installed on your Unix computer.

Note that only TCP/IP-RFC1006 is present by default on Unix systems.

- Entry of a partner address (openFT for BS2000/OSD partners) 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 dress for transfer via TCP/IP-RFC1006 and a PCMX, CMX-V4.0 or Windows partner (as of FT-PCD V2.6):

```
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
```

- Entry of a partner address for transfer via ETHN-CLNS/active:

```
ftethna\
        TA      OSITYPE 49+006C080015304050FE T'$FJAM'
 ;                      OSI network addr. T selector
```

(OSI network address as per ISO Standard 8348/Add.2, the structure is described in the CMX manual.)

- Entry of a partner address for transfer via ETHN-CLNS/passive:

```
ftethnp\
        TA      LANSBKA 080014110960  T'$FJAM'
 ;                      Ethernet addr. T selector
```

- Entry of a partner address for transfer via WAN-NEA, WAN-NX25, ISDN-NEA, ISDN-NX25

```
ftwannea\
        TA      WANNEA T'$FJAM'    1/18            WAN 2
;                     T selector  Proc./region    WAN CC
```

- Entry of a partner address for transfer via WAN-CONS, ISDN-CONS

```
ftcons\
        TA     WANSBKA X.121 45890012233 T'$FJAM'  WAN 3
 ;                       SNPA info       T Sel.    WAN CC
```

9.3.4 Definition of remote TS applications for openFT-FTAM

In the case of FTAM partners, TNS entries are only necessary if these partners are not connected via TCP/IP. 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, no TNS entries are required any longer as of openFT V10 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, the entries `PSEL V''` or `SSEL V''` are absolutely necessary. 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

The entries of the file to be translated with *tnsxcom* must in principle look the same as in the following examples on page 427. 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.

1. openFT-FTAM sets up the connection.

   Which values do the following parameter have (with specification of coding):

a)	called X121/ LAN address/ NSAP/X.31	_____		
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 X121/ LAN address/ NSAP/X.31	_____		
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.

9.3.4.1 Sample entries for FTAM partners

The examples listed below assume that the corresponding transport system is installed on your Unix computer.

Note that only TCP/IP-RFC1006 is present by default on Unix systems.

- 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 ≤ V10.0 for Windows 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
```

- Entry of a partner address for transfer via ETHN-CLNS/active:

```
ftametha\
    PSEL    V''
    SSEL    V''
    TA      OSITYPE 49+006C080015304050FE T'$FTAM'
 ;                        OSI network addr. T selector
```

(OSI network address as per ISO Standard 8348/Add.2; the structure is described in the CMX manual.)

- Entry of a partner address for transfer via ETHN-CLNS/passive:

```
ftamethp\
    PSEL    V''
    SSEL    V''
    TA      LANSBKA 080014110960  T'$FTAM'
 ;                 Ethernet addr.T selector
```

- Entry of a partner address for transfer via WAN-CONS, ISDN-CONS

```
ftamcons\
    PSEL    V''
    SSEL    V''
    TA      WANSBKA X.121 45890040034 T'$FTAM' X'D5000002'
 ;                        SNPA info    T sel.   TPI
```

9.4 openFT in a Cluster with Unix based systems

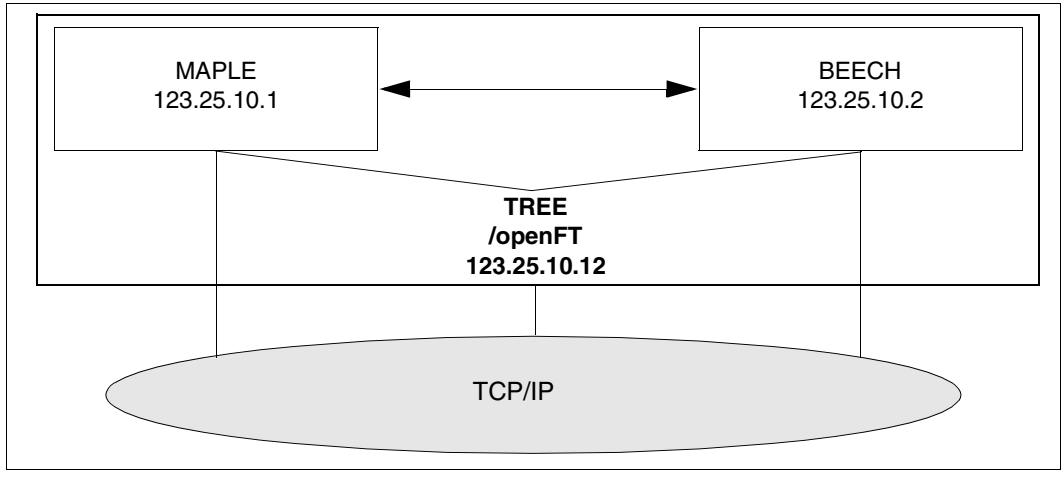
Software requirements

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” on page 436.

You are recommended to work without CMX and TNS.

9.4.1 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:

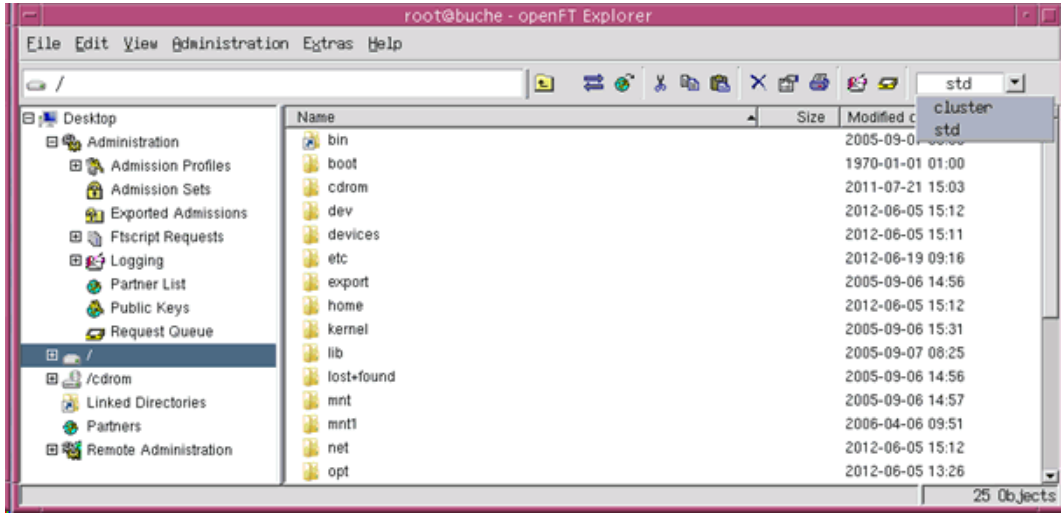
   ```
   PAR=$1
   BIN=/opt/bin; export BIN
   INST=cluster
   OPENFTINSTANCE=$INST; export OPENFTINSTANCE
   case $PAR in
   start) OPENFTINSTANCE=std; export OPENFTINSTANCE
       $BIN/ftcrei $INST /openFT/$INST
       case $? in
          0|5) continue;;
          *) exit 1;;
       esac
       OPENFTINSTANCE=$INST; export OPENFTINSTANCE
       $BIN/ftstart 2>/dev/null
       case $? in
          0|180) exit 0;;
          *) exit 1;;
       esac;;
   ```

```
stop) $BIN/fttop 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 [ "$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 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.

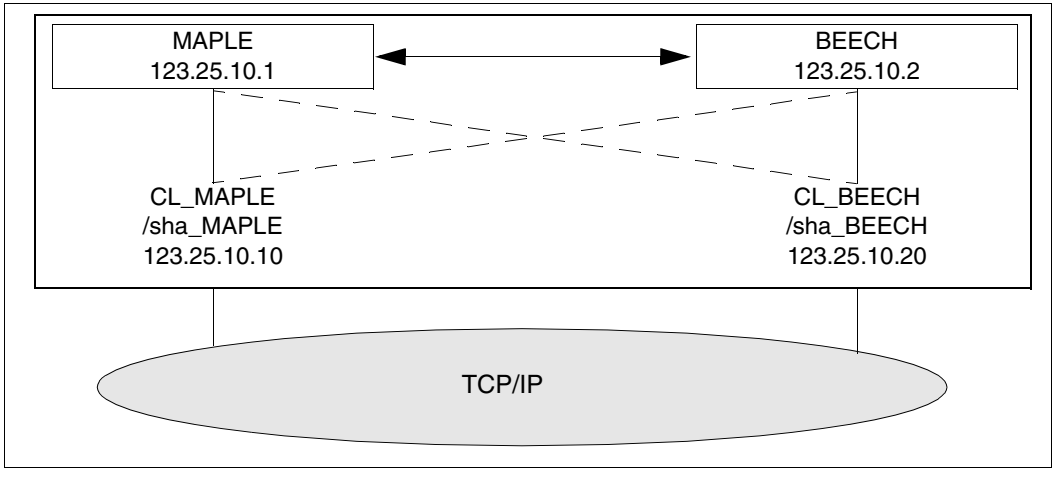


Selecting an instance in a cluster

9.4.2 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_MEAPLE/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 $? in
      0)exit 0;;
      1)exit1;;
   esac;;
check) VALUE=`$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.

9.4.3 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*.

9.5 Exit codes and messages for administration commands

Below is a description of the error messages output by openFT together with the associated exit codes, meanings and measures as appropriate.

The description has the following format:

exit code Message text
meanings and measures as appropriate

9.5.1 Messages for all commands

0 The command was successful

3 The command was cancelled as the result of a response to a query

4 A syntax error occurred during command processing

225 Information output canceled

Meaning:
A show command was interrupted, for example.

Measure:
Repeat the command.

226 Monitor file contents inconsistent

Meaning:
The command cannot be accepted because the contents of the specified monitor file are inconsistent.
Possible reason: The monitor file was accessed by the user in a mode other than read mode while it was monitoring an FT request.
The contents of the monitor file can no longer be used.

227 Monitor file not in use by openFT

Measure:
Correct the name of the job variable and repeat the command.

228 Monitor file not found

Measure:
Correct the name of the job variable and repeat the command.

236 Current instance '<instance>' no longer found

Meaning:
The command was rejected. The instance '<instance>' could not be found.

250 An internal error occurred during command processing

251 Command aborted with core dump

253 Current openFT instance is invalid

Meaning:
During command processing a defined instance was found to be invalid

255 ftexec/ftadm command failed

Meaning:
Remote execution of the command with ftexec failed

9.5.2 Messages for administration commands and measurement data recording

With the following messages, the value for *fthelp* must be increased by 1000, e.g. 1034 instead of 34.

20 openFT already started

Meaning:
openFT can only be started once in each instance.

Measure:
Terminate openFT if necessary.

21 Request must be canceled without FORCE option first

Meaning:
Before the FORCE option is used, the command must be called without the FORCE option.

Measure:
Issue the command without the FORCE option first.

29 Maximum number of key pairs exceeded

Measure:
Before new key pair set can be created, an older key pair set must be deleted.

30 Warning: last key pair deleted

Meaning:
The last key pair set has been deleted. Without a key pair set, encrypted transfer, authentication and data integrity checking are not possible.

Measure:
Create a new key pair set.

31 No key pair available

Meaning:
All transfers are carried out without encryption.

Measure:
Create a new key pair set, if necessary.

32 Last key pair must not be deleted

33 The public key files could not be updated

Meaning:
The contents of the *syspkf* file could not be fully updated.
Possible reasons:
– The *syspkf* file is locked.
– There is not enough disk space to allow the file to be created.

Measure:
Take the appropriate action depending on the cause of the error:
– Unlock the file.
– Allocate disk space or have your system administrator do it.

Update the key with *ftupdk*.

34 Command only permissible for FT, FTAC or ADM administrator

Meaning:
Only the FT, FTAC or ADM administrator is permitted to use the command.

Measure:
Have the command executed by the FT, FTAC or ADM administrator.

35 Command only permissible for FT administrator

Meaning:
Only the FT administrator is permitted to use the command.

Measure:
Have the command executed by the FT administrator.

36 User not authorized for other user Ids

Meaning:
The user is not authorized to use a different user ID in the command.

Measure:
Specify your own ID, or have the command executed by the FT or FTAC administrator.

37 Key reference unknown

Meaning:
The specified key reference is unknown.

Measure:
Repeat the command with an existing key reference.

38 Request <Request id> is in the termination phase and can no longer be canceled

39 openFT not active

Meaning:
openFT is not started.

Measure:
Start openFT, if necessary.

40 Config user ID unknown or not enough space

Meaning:
The Config user ID of the current instance is unknown or the disk space allocated is insufficient to allow creation of the request file, the file for storing trace data, or the key files.

Measure:
Either create the Config user ID or increase its disk space allocation or have your system administrator do it.

41 Specified file is not a valid trace file

42 openFT could not be started

43 Partner with same attribute <attribute> already exists in partner list

Meaning:
There is already a partner entry with the same attribute <attribute> in the partner list.

Measure:
The attribute <attribute> in partner entries must be unique. Correct the command accordingly and try again.

44 Maximum number of partners exceeded

Meaning:
The partner list already contains the maximum permissible number of partner entries.

Measure:
Delete partners that are no longer required.

45 No partner found in partner list

Meaning:
A partner for the specified selection could not be found in the partner list.

Measure:
Check if the specified partner name or address was correct.
If necessary, repeat the command using the correct name or address.

46 Modification of partner protocol type not possible

Meaning:
The protocol type of the partner entry cannot be changed subsequently.

Measure:
Delete the partner from the partner list, if necessary, and enter it again with a new protocol type.

47 Request <Request id> not found

Meaning:
The request with the transfer ID <Request id> could not be found.

Measure:
Specify the existing transfer ID and repeat the command.

48 Active requests could not yet be deleted

49 CCS name '<1>' unknown

50 ftscript process could not be started

51 Error displaying an ftscript user

52 ftscript user number limit exceeded

53 ftscript chapter not found

54 ftscript id not found

55 ftscript file not found

56 ftscript request is still running

57 Inbound requests cannot be modified

58 The ADM trap server configuration is invalid

59 monitoring is not active

Meaning:
The command is only supported if monitoring is activated.

Measure:
Activate monitoring in the operating parameters and repeat the command.

60 File could not be created

Meaning:
The command was not executed because the local file could not be created.

Measure:
Check the directory and access rights. Repeat the command.

61 Higher-level directory not found

Meaning:
The local file could not be created when exporting the configuration data because the specified path does not exist.

Measure:
Create or correct the path for the configuration file and repeat the command.

62 File already exists

Meaning:
The command was not executed because the specified file already exists.

Measure:
Either delete the existing configuration file or choose a different name and repeat the command.

63 Resulting file name too long

Meaning:
The filename has the wrong syntax or is too long. Specifying a partially qualified filename may be the cause of the error.

Measure:
Repeat the command using the correct syntax.

64 File locked to prevent multiple access

Meaning:
The command was not executed because the file is already locked by another process.

Measure:
Repeat the command later.

65 File not found

Meaning:
The command was not executed because the specified file was not found.

Measure:
Correct the file name and repeat the command.

66 Not enough space for file

Meaning:
The command was not executed because the permitted storage space on the local volume is exhausted.

Measure:
Take appropriate measures depending on the cause of the error.

– Delete any files that are no longer required or

– Request the system administrator to assign more storage space.

67 Syntax error in resulting file name

Meaning:
The file cannot be accessed because the absolute file name has become too long, for instance.

Measure:
Shorten the path or the file name. Repeat the command.

68 Access to file denied<2>

Meaning:
The command was not executed because the file only permits certain access modes (e.g. read-only).

Measure:
Correct the file name or the file protection attributes.
Repeat the command.

69 Error accessing file<2>

Meaning:
<2>: DMS error

Measure:
Take appropriate measures depending on the error code.

70 Configuration data invalid

Meaning:
The configuration data is syntactically or semantically incorrect and can therefore not be imported.

Measure:
Correct the error on the basis of the additional diagnostic output and then repeat import of the configuration data.

71 Import of configuration data not possible while remote administration server is started

Meaning:
The changes to the configuration data are so extensive that they can only be imported when the remote administration server has been terminated.

Measure:
Terminate openFT using the *ftstop* command and then attempt to import the configuration data again.

73 Command aborted

Meaning:
The user has cancelled the command.

74 Command only permissible for ADM administrator on a remote administration server

Meaning:
The command is only permitted for the ADM administrator.

Measure:
Have the ADM administrator execute the command if necessary.

77 Not possible to change the transport access system. Reason: <1>

Meaning:
The operating mode with or without CMX could not be changed using the *ftmodo* command. This may be due to the following causes:

openFT has been started

CMX is not installed

78 Interval too short since last change of log file

Meaning:
Log file cannot currently be changed because the timestamp-dependent part of the file name is the same as that of the current log file.

Measure:
Wait for a short time (if necessary) and repeat the command.

9.5.3 Messages for remote administration

With the following messages, the value for *fthelp* must be increased by 2000, e.g. 2052 instead of 52.

52 Administration request rejected by remote administration server

Meaning:
The administration request was rejected by the remote administration server because it clashes with the settings in the configuration file of the remote administration server.

The ADM administrator can determine the precise reason for rejection from the associated ADM log record on the remote administration server.

Possible reason codes:

7001 The administrator ID is invalid. It was not possible to determine a valid administrator ID from the user ID or the profile name in the configuration data of the remote administration server.

7002 The routing information is invalid. The specified openFT instance specified in the routing information could not be found in the configuration data of the remote administration server.

7003 The specified remote administration command is invalid. The remote administration server rejects the specified command because it is not a supported remote administration command.

7101 Infringement against the access rights list. On checking the access rights, the system identified that the administrator ID has not been assigned the necessary rights in the configuration data of the remote administration server to be able to execute the valid remote administration command on the specified openFT instance.

7201 Infringement against the maximum command length. In particular in the case of BS2000 commands, the remote administration server replaces the shortest command names, which are guaranteed by openFT, by the full command names. If this replacement of the command name causes the entire remote administration command to become longer than the maximum command length of 8192 characters, the command is rejected.

Measure:
Have the ADM administrator carry out the necessary adjustments to the configuration data or check the command. Repeat the changed command if necessary.

54 Invalid command

Meaning:
The specified command is not a command that is permitted to be executed on the specified system using the remote administration facility.

Measure:
Specifying an admissible command or add the missing routing information. Repeat the command.

57 openFT is not authorized to execute administration requests

Meaning:
openFT is not (no longer) authorized to process administration requests. This is, for example, the case if a remote administration server has been demoted to a normal server (*ftmodo -admcs=n*) or if commands that are only allowed to be executed on a remote administration server are processed by an openFT instance that has not been configured as a remote administration server.

Glossary

Italic type indicates a reference to other terms in this glossary.

absolute path name
The entire path name, from the root directory to the file itself.

access control
File attribute in the *virtual filestore*, attribute of the *security group* that defines *access rights*.

access protection
Comprises all the methods used to protect a data processing system against unauthorized system access.

access right
Derived from the *transfer admission*. The access right defines the scope of access for the user who specifies the transfer admission.

action list
Component of the file attribute *access control* (attribute of the *security group*) in the *virtual filestore* that defines *access rights*.

ADM administrator
Administrator of the *remote administration server*. This is the only person permitted to modify the configuration data of the remote administration server.

ADM partner
Partner system of an openFT instance with which communication takes place over the *FTADM protocol* in order to perform *remote administration*.

ADM traps
Short messages sent to the *ADM trap server* if certain events occur during operation of openFT.

ADM trap server
Server that receives and permanently stores the *ADM traps*. It must be configured as a *remote administration server*.

administrated openFT instance

openFT instances that are able to be administered by *remote administrators* during live operation.

admission profile

Way of defining the *FTAC* protection functions. Admission profiles define a *transfer admission* that has to be specified in *FT requests* instead of the *LOGON* or *Login authorization*. The admission profile defines the *access rights* for a user ID by restricting the use of parameters in *FT requests*.

admission profile, privileged

see *privileged admission profile*

admission set

In *FTAC*, the admission set for a particular user ID defines which FT functions the user ID may use and for which *partner systems*.

admission set, privileged

see *privileged admission set*

AES (Advanced Encryption Standard)

The current symmetrical encryption standard, established by NIST (National Institute of Standards and Technology), based on the Rijndael algorithm, developed at the University of Leuven (B). The openFT product family uses the AES method to encrypt the request description data and possibly also the file contents.

ANSI code

Standardized 8-bit character code for message exchange. The acronym stands for "American National Standards Institute".

API (Application Programming Interface)

An interface that is freely available to application programmers. It provides a set of interface mechanisms designed to support specific functionalities.

Application Entity Title (AET)

The Application Entity Title consists of Layer 7 addressing information of the *OSI Reference Model*. It is only significant for *FTAM partners*.

asynchronous request

Once the *FT request* has been submitted, it is processed independently of the user. The user can continue working once the system has confirmed acceptance of the request. (see also *synchronous request*).

authentication

Process used by openFT to check the unique identity of the request partner.

basic functions

Most important file transfer functions. Several basic functions are defined in the *admission set* which can be used by a login name. The six basic functions are:

– inbound receive
– inbound send
– inbound follow-up processing
– inbound file management
– outbound receive
– outbound send

central administration

Central administration in openFT incorporates the *remote administration* and *ADM traps* functions and requires the use of a *remote administration server*.

character repertoire

Character set of a file in the *virtual filestore.*
In the case of files transferred with *FTAM partners* it is possible to choose between: *GeneralString*, *GraphicString*, *IA5String* and *VisibleString*.

Character Separated Values (CSV)

This is a quasi-tabular output format that is very widely used in the PC environment in which the individual fields are separated by a separator (often a semicolon ";"). It permits the further processing of the output from the most important openFT commands using separate tools.

client

– Term derived from client/server architectures: the partner that makes use of the services provided by a *server*.
– Logical instance which submits requests to a *server.*

cluster

A number of computers connected over a fast network and which in many cases can be seen as a single computer externally. The objective of clustering is generally to increase the computing capacity or availability in comparison with a single computer.

Comma Separated Values

see *Character Separated Values.*

communication controller

see *preprocessor*

compression

This means that several identical successive characters can be reduced to one character and the number of characters is added to this. This reduces transfer times.

computer network, open

see *open computer network*

Component of the FTAM file attribute *access control* (part of the *security group*) in the *virtual filestore* that controls concurrent access.

connectivity

In general, the ability of systems and partners to communicate with one another. Sometimes refers simply to the communication possibilities between transport systems.

constraint set

Component of the *document type*.

contents type

File attribute in the *virtual filestore,* attribute of the *kernel group* that describes the file structure and the form of the file contents.

data communication system

Sum of the hardware and software mechanisms which allow two or more communication partners to exchange data while adhering to specific rules.

data compression

Reducing the amount of data by means of compressed representation.

data encoding

Way in which an *FT system* represents characters internally.

Data Encryption Standard (DES)

International data encryption standard for improved security. The DES procedure is used in the FT products to encrypt the request description data and possibly the request data if connections are established to older versions of openFT that do not support *AES*.

data protection

– In the narrow sense as laid down by law, the task of protecting personal data against misuse during processing in order to prevent the disclosure or misappropriation of personal information.
– In the wider sense, the task of protecting data throughout the various stages of processing in order to prevent the disclosure or misappropriation of information relating to oneself or third parties.

data security

Technical and organizational task responsible for guaranteeing the security of data stores and data processing sequences, intended in particular to ensure that

– only authorized personnel can access the data,
– no undesired or unauthorized processing of the data is performed,
– the data is not tampered with during processing,
– the data is reproducible.

DHCP

Service in TCP/IP networks that automatically assigns IP addresses and TCP/IP parameters to clients on request.

directory

Directories are folders in the hierarchical file system of a Unix system (including POSIX) or a Windows system that can contain files and/or further directories.

document type

Value of the file attribute *contents type* (attribute of the *kernel group*). Describes the type of file contents in the *virtual filestore*.

– *document type* for text files: FTAM-1
– *document type* for binary files: FTAM-3

dynamic partner

partner system that is either not entered in the *partner list* (*free dynamic partner*) or that is entered in the partner list with only address but without a name (*registered dynamic partner*).

EBCDIC

Standardized code for message exchange as used in BS2000/OSD. The acronym stands for "Extended Binary Coded Decimal Interchange Code".

emulation

Components that mimic the properties of another device.

entity

see *instance*

Explorer

A program from Microsoft that is supplied with Windows operating systems to facilitate navigation within the file system.

file attributes

A file's properties, for example the size of the file, access rights to the file or the file's record structure.

file management

Possibility of managing files in the remote system. The following actions are possible:

- Create directories
- Display and modify directories
- Delete directories
- Display and modify file attributes
- Rename files
- Delete files.

file transfer request

see *FT- request*

firewall processor

Processor which connects two networks. The possible access can be controlled precisely and also logged.

fixed-length record

A record in a file all of whose records possess the same, agreed length. It is not necessary to indicate this length within the file.

follow-up processing

FT function that initiates execution of user-specified commands or statements in the *local* and/or the *remote system* after an *FT request* has been completed. The user may define different follow-up processing, depending on the success or failure of FT request processing. See also *preprocessing* and *postprocessing*.

follow-up processing request

Statements contained within an *FT request* which perform *follow-up processing* after file transfer.

free dynamic partner

Partner system that is not entered in the partner list.

FT administrator

Person who administers the openFT product installed on a computer.

FT request

Request to an *FT system* to transfer a file from a *sending system* to a *receive system* and (optionally) start *follow-up processing requests*.

FT system

System for transferring files that consists of a computer and the software required for file transfer.

FT trace

Diagnostic function that logs FT operation.

FTAC (File Transfer Access Control)

Extended access control for file transfer and file management. In the case of BS2000 and z/OS, this is implemented by means of the product openFT-AC, for other operating systems it is a component of the openFT product, e.g. in openFT for Unix systems or openFT for Windows systems.

FTAC administrator

Administrator of the FTAC functions; should be identical to the person responsible for data security in the system.

FTAC logging function

Function which FTAC uses to log each access to the protected system via file transfer.

FTADM protocol

Protocol used for communication between two openFT instances in order to perform *remote administration* or transfer *ADM traps*.

FTAM-1

document type for text files

FTAM-3

document type for binary files

FTAM catalog

The FTAM catalog is used to extend the file attributes available in Unix systems. It is only relevant for access using FTAM. For example, a file can be deleted using the command *erase* on a Windows system, even if the *permitted actions* parameter does not allow this.

FTAM file attributes

All systems which permit file transfer via FTAM protocols must make their files available to their partners using a standardized description (ISO 8571). To this end, the attributes of a file are mapped from the physical filestore to a *virtual filestore* and vice versa. This process distinguishes between three groups of file attributes:

– kernel group: describes the most important file attributes.
– storage group: contains the file's storage attributes.
– security group: defines security attributes for file and system access control.

FTAM partner

Partner system that uses *FTAM protocols* for communication.

FTAM protocol (File Transfer, Access and Management)

Protocol for file transfer standardized by the "International Organization for Standardization" (ISO) (ISO 8571, FTAM).

FTP partner

Partner system that uses *FTAM protocols* for communication.

FTP protocol

Manufacturer-independent protocol for file transfer in TCP/IP networks.

functional standard

Recommendation defining the conditions and the forms of application for specific ISO standards (equivalent term: *profile*). The transfer of unstructured files is defined in the European Prestandard CEN/CENELEC ENV 41 204; file management is defined in the European Prestandard CEN/CENELEC ENV 41205.

gateway

Generally understood to mean a computer that connects two or more networks and which does not function as a bridge. Variants: gateway at network level (= router or OSI relay), transport and application gateway.

gateway processor

Communication computer that links a computer network to another computer network. The mapping of the different protocols of the various computer networks takes place in gateway processors.

general string
Character repertoire for file files transferred to and from *FTAM partners*.
global request identification / global request ID Request number that the *initiator* of an openFT or FTAM request transfers to the *responder*. This means that the global request ID in the responder is identical to the *request ID* in the initiator. The responder generates its own (local) request ID for the request. This means that information stored in both the initiator and the responder can be unambiguously assigned to a request. This is particularly important if the request has to be restarted.

GraphicString
Character repertoire for files transferred to and from *FTAM partners*.

heterogeneous network
A network consisting of multiple subnetworks functioning on the basis of different technical principles.

homogeneous network
A network constructed on the basis of a single technical principle.

HOSTS file
Network administration file that contains the Internet addresses, the processor names and the alias names of all accessible computers.

IA5String
Character repertoire for files transferred to and from *FTAM partners.*

identification
Procedure making it possible to identify a person or object.

inbound file management
Request issued in a *remote system* for which directories or file attributes of the *local system* can be displayed, file attribute modified or local file deleted.

inbound follow-up processing
Request issued in a *remote system* with *follow-up processing* in the *local system*.

inbound receive
Request issued in the *remote system*, for which a file is received in the *local system*.

inbound request / inbound submission
Request issued in another system, i.e. for this request.

inbound send

Request issued in a *remote system* for which a file is sent from the *local system* to the remote system.

initiator

Here: *FT system* that submits an *FT request*.

instance / entity

A concept of OSI architecture: active element in a layer. Also see *openFT instance*.

instance ID

A network-wide, unique address of an openFT instance.

integrity

Unfalsified, correct data following the processing, transfer and storage phases.

interoperability

Capability of two *FT systems* to work together.

ISO/OSI reference model

The ISO/OSI Reference Model is a framework for the standardization of communications between open systems. (ISO=International Standards Organization).

job

Sequence of commands, statements and data.

job transfer

Transfer of a file that constitutes a *job* in the *receive system* and is initiated as a job there.

kernel group

Group of file attributes of the *virtual filestore* that encompasses the kernel attributes of a file.

library

File with internal structure (elements)

library element

Part of a library. A library may in turn be subdivided into a number of records.

Local Area Network (LAN)
Originally a high-speed network with limited physical extension. Nowadays, any network, that uses CSMA/CD, Token Ring or FDDI irrespective of the range (see also *WAN Wide Area Network*).

local system
The *FT system* at which the user is working.

logging function
Function used by openFT to log all file transfer accesses to the protected system.

log record
Contains information about access checks performed by openFT (FTAC log record) or about a file transfer or remote administration request which is started when the access check was successful (FT log record or ADM log record).

Logical Unit (LU)
Interface between an application program and the SNA data communications network. The LU type describes the communications characteristics.

Login authorization
Transfer admission to a computer which (as a rule) consists of the login name and the password, and authorizes dialog operation, see also *LOGON authorization*.

LOGON authorization
Transfer admission authorizing access to a computer. The LOGON authorization (normally) consists of user ID, account number and password and authorizes the user to make use of interactive operation.

named partner
partner system entered by its name in the *partner list*.

Network Control Program (NCP)
Operating system of the front-end-processor for SNA hosts.

network description file
File used up to openFT V9 that contains specifications concerning *remote systems* (*FT systems*).

offline logging

The log file can be changed during operation. Following this changeover, the previous log file is retained as an offline log file; new log records are written to a new log file. It is still possible to view the log records in an offline log file using the tools provided by openFT.

open computer network

Computer network in which communication is governed by the rules of ISO/OSI. Interoperation of different computers from various vendors is made possible by defined *protocols*.

openFT Explorer

openFT program that provides a graphical user interface that allows file transfer and administration functions to be performed.

openFT installation directory

Path under which openFT is installed. This path can be freely selected during interactive installation. It can be set with the INSTALLDIR parameter during unattended installation. The default path depends on the language setting and the version of the Windows operating system.
(Default: %Program Files%\openFT).

openFT instance

Several openFT systems, so-called openFT instances, can be running simultaneously on an individual computer or a cluster of a TCP/IP network. Each instance has its own address (instance ID) and is comprised of the loaded code of the openFT products (including add-on products if they are available) and of the variable files such as partner list, logging files, request queue, etc.

openFT Monitor

Program that allows the monitoring data for openFT operation to be shown in the form of a chart. openFT Monitor requires a graphics-capable terminal.

openFT partner

Partner system which is communicated with using *openFT protocols*.

openFT protocols

Standardized *protocols* for file transfer (SN77309, SN77312).

openFT-FTAM

Add-on product for openFT (for BS2000, Unix systems and Windows systems) that supports file transfer using FTAM protocols. FTAM stands for File Transfer, Access and Management (ISO 8571).

openFT-Script
: openFT interface providing an XML based script language that includes file transfer and file management functions. This interface allows you to combine several file transfer or file management requests to form a single openFT-Script request.

openFT-Script commands
: Commands used for administering openFT-Script requests.

operating parameters
: Parameters that control the *resources* (e.g. the permissible number of connections).

outbound request / outbound submission
: Request issued in your own processor.

outbound receive
: Request issued locally for which a file is received in the *local system*.

outbound send
: Request issued locally for which a file is sent from the *local system*.

owner of an FT request
: Login name in the *local system* or *remote system* under which this *FT request* is executed. The owner is always the ID under which the request is submitted, not the ID under which it is executed.

partner
: see *partner system*

partner list
: File containing specifications concerning *remote systems* (*FT systems*).

partner system
: Here: *FT system* that carries out *FT requests* in cooperation with the *local system*.

password
: Sequence of characters that a user must enter in order to access a user ID, file, job variable, network node or application. The user ID password serves for user *authentication*. It is used for access control. The file password is used to check access rights when users access a file (or job variable). It is used for file protection purposes.

permitted actions
File attribute in the *virtual filestore*; attribute of the *kernel group* that defines actions that are permitted in principle.

port number
Number that uniquely identifies a TCP/IP application or the end point of a TCP/IP connection within a processor.

POSIX (Portable Open System Interface)
Board and standards laid down by it for interfaces that can be ported to different system platforms.

postprocessing
openFT makes it possible to process the received data in the receiving system through a series of operating system commands. Postprocessing runs under the process control of openFT (in contrast to *follow-up processing*).

preprocessing
The preprocessing facility in openFT can be used to send a receive request in which the outputs of a remote command or program are transferred instead of a file. This makes it possible to query a database on a remote system, for example. Preprocessing also may be issued locally.

presentation
Entity that implements the presentation layer (layer 6) of the *ISO/OSI Reference Model* in an *FT system* that uses e.g. *FTAM protocols*.

presentation selector
Subaddress used to address a *presentation application*.

private key
Secret decryption key used by the recipient to decrypt a message that was encrypted using a *public key*. Used by a variety of encryption procedures including the *RSA procedure*.

privileged admission profile
Admission profile that allows the user to exceed the *FTAC administrator's* presettings in the *admission set*. This must be approved by the *FTAC administrator* who is the only person able to privilege admission profiles.

privileged admission set
Admission set belonging to the *FTAC administrator*.

profile

In OSI, a profile is a standard which defines which protocols may be used for any given purpose and specifies the required values of parameters and options. Here: a set of commands assigned to a user ID. The permissibility of these commands is ensured by means of syntax files. See also *admission profile*, *privileged admission profile*.

prompting in procedures

Function used to prompt the user at the terminal to enter data required to run the procedure.

protocol

Set of rules governing information exchange between peer partners in order to achieve a defined objective. This usually consists of a definition of the messages that are to be exchanged and the correct sequencing of messages including the handling of errors and other exceptions.

public key

Public encryption key defined by the receiver of a message, and made public or made known to the sender of the message. This allows the sender to encrypt messages to be sent to the receiver. Public keys are used by various encryption methods, including the *Rivest Shamir Adleman (RSA) procedure*. The public key must match the *private key* known only to the receiver.

receive file

File in the *receive system* in which the data from the *send file* is stored.

receive system

System to which a file is sent. This may be the *local system* or the *remote system*.

record

Set of data that is treated as a single logical unit.

registered dynamic partner

Partner system that is entered in the partner list with only an address but no name.

relative path name

The path from the current *directory* to the file.

remote administration

Administration of openFT instances from remote computers.

remote administration server
Central component required for *remote administration* and for *ADM traps*. A remote administration server runs on a Unix or Windows system running openFT as of V11.0. If it is used for *remote administration*, it contains all the configuration data required for this purpose.

remote administrator
Role configured on the *remote administration server* and which grants permission to execute certain administration functions on certain openFT instances.

remote system
see *partner system*

request
see *FT request*

request queue
File containing *asynchronous requests* and their processing statuses.

request identification / request ID
Number assigned by the local system that identifies an *FT request.*

request management
FT function responsible for managing *FT requests*; it ensures request processing from the submission of a request until its complete processing or termination.

request number
see *request identification*

request storage
FT function responsible for storing *FT requests* until they have been fully processed or terminated.

resources
Hardware and software components needed by the *FT system* to execute an *FT request* (processes, connections, lines). These resources are controlled by the *operating parameters*.

responder
Here: *FT system* addressed by the *initiator*.

restart
Automatic continuation of an *FT request* following an interruption.

restart point

Point up to which the data of the *send file* has been stored in the *receive file* when a file transfer is interrupted and at which the transfer of data is resumed following a *restart*.

result list

List with information on a completed file transfer. This is supplied to the user in the *local system* and contains information on his or her *FT requests*.

RFC (Request for Comments)

Procedure used on the Internet for commenting on proposed standards, definitions or reports. Also used to designate a document approved in this way.

RFC1006

Supplementary protocol for the implementation of ISO transport services (transport class 0) using TCP/IP.

Rivest-Shamir-Adleman-procedure (RSA procedure)

Encryption procedure named after its inventors that operates with a key pair consisting of a *public key* and a *private key*. Used by the openFT product family in order to reliably check the identity of the partner system and to transmit the AES key to the partner system for encrypting the file contents.

router

Network element that is located between networks and guides message flows through the networks while simultaneously performing route selection, addressing and other functions. Operates on layer 3 of the OSI model.

security attributes

An object's security attributes specify how and in what ways the object may be accessed.

Secure FTP

Method by which a connection is tunneled using the *FTP protocol*, thus allowing secure connections with encryption and *authentication*.

security group

Group of file attributes in the *virtual filestore*, encompassing the security attributes of a file.

security level

When *FTAC functions* are used, the security level indicates the required level of protection against a *partner system*.

send file

File in the *sending system* from which data is transferred to the *receive file*.

sending system

Here: *FT system* that sends a file. This may be the *local system* or the *remote system*.

server

Logical entity or application component which executes a client's requests and assures the (coordinated) usage of all the generally available services (File, Print, data base, Communication, etc.). May itself be the client of another server.

service

- As used in the OSI architecture: a service is the set of functions that a service provider makes available at a service access point.
- As used in the client/server architecture: a set of functions that a server makes available to its clients.
- Term used in Unix and Windows systems: A program, routine or process used to perform a particular system function to support other programs, in particular on a low level (hardware-related).

service class

Parameter used by *FTAM partners* to negotiate the functions to be used.

session

- In OSI, the term used for a layer 5 connection.
- In SNA, a general term for a connection between communication partners (applications, devices or users).

session selector

Subaddress used to address a *session* application.

shell metacharacters

The following metacharacters have special meanings for the shell (= Windows command prompt): *, [ ], ?, <, >, |, &, &&, ( ), { }

SNA network

Data communication system that implements the Systems Network Architecture (SNA) of IBM.

SNMP (Simple Network Management Protocol)

Protocol for TCP/IP networks defined by the Internet Engineering Task Force (IETF) for the transfer of management information.

special characters
see *shell metacharacters.*

standard admission set
This standard admission set applies by default to all users for whom there is no dedicated admission set. These default settings may be restricted further by the user for his or her own admission set.

standard error output (stderr)
By default, standard error output is to the screen.

standard input (stdin)
By default, standard input is from the keyboard.

standard output (stdout)
By default, standard output is to the screen.

storage group
File attribute in the *virtual filestore*, encompasses the storage attributes of a file.

string
Character string

string significance
Describes the format of *strings* in files to be transferred using *FTAM protocols*.

synchronous request
The user that submitted the *FT request* waits for transfer to terminate. The user cannot continue working (see also *asynchronous request*).

system
see *FT- system*

system, local
see *local system*

system, remote
see *remote system*

TCP/IP (Transmission Control Protocol / Internet Protocol)
Widely used data transmission protocol (corresponds approximately to layers 3 and 4 of the *ISO/OSI reference model*, i.e. network and transport layers); originally developed for the ARPANET (computer network of the US Ministry of Defense) it has now become a de-facto standard.

transfer admission

Authorization for file transfer and file management when using FTAC. The transfer admissions is then used in place of the *LOGON or LOGIN authorization*.

transfer unit

In an FTAM environment, the smallest data unit for transporting file contents. For *FTAM-1* and *FTAM-3* these are *strings*. A transfer unit can, but need not, correspond to one file record.

Transmission Control Protocol / Internet Protocol

see *TCP/IP*

TranSON

TranSON is a software product that permits secure access to a server. The use of TranSON is transparent to the application. The connection to the remote partner goes from the workstation through a client proxy and server proxy to the remote partner. The client proxy is located on the workstation, and the server proxy is located on the remote partner. The data transferred between the client proxy and the server proxy is encrypted.

transport connection

Logical connection between two users of the transport system (terminals or applications).

transport layer

Layer 4 of the *ISO/OSI reference model* on which the data transport protocols are handled.

Transport Name Service (TNS)

Service used to administer properties specific to transport systems. Entries for *partner systems* receive the information on the particular *transport system* employed.

transport protocol

Protocol used on the *transport layer*

transport selector (T-selector)

Subaddress used to address an ISO-8072 application in the *transport layer*.

transport system

- The part of a system or architecture that performs approximately the functions of the four lower OSI layers, i.e. the transport of messages between the two partners in a communication connection.
- Sum of the hardware and software mechanisms that allow data to be transported in computer networks.

Unicode

The universal character encoding, maintained by the Unicode Consortium. This encoding standard provides the basis for processing, storage and interchange of text data in any language in all modern software and information technology protocols. The Unicode Standard defines three Unicode encoding forms: UTF-8, UTF-16 and UTF-32.

universal-class-number

Character repertoire of a file in the *virtual filestore.*

UNIX®

Registered trademark of the Open Group for a widespread multiuser operating system. A system may only bear the name UNIX if it has been certified by the Open Group.

Unix system

Commonly used designation for an operating system that implements functions typical of UNIX® and provides corresponding interfaces. POSIX and Linux are also regarded as Unix systems.

variable length record

A record in a file all of whose records may be of different lengths. The record length must either be specified in a record length field at the start of the record or must be implicitly distinguishable from the next record through the use of a separator (e.g. Carriage Return - Line Feed).

virtual filestore

The FTAM virtual filestore is used by *FT systems* acting as *responders* to make their files available to their *partner systems*. The way a file is represented in the virtual filestore is defined in the FTAM standard, see *file attributes*.

VisibleString

Character repertoire for files transferred to and from *FTAM partners*.

WAN (Wide Area Network)

A public or private network that can span large distances but which runs relatively slowly and with higher error rates when compared to a *LAN*. Nowadays, these definitions have only limited validity. Example: in ATM networks.

Abbreviations

ACSE	Association Control Service Element
AES	Advanced Encryption Standard
AET	Application Entity Title
ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
BCAM	Basic Communication Access Method
CAE	Common Application Environment
CCP	Communication Control Programm
CCS	Coded Character Set
CEN	Comité Européen de Normalisation
CENELEC	Comité Européen de Normalisation Électrotechnique
CSV	Character Separated Values
CMX	Communication Manager Unix Systems
DCAM	Data Communication Access Method
DCM	Data Communication Method
DES	Data Encryption Standard
DIN	Deutsches Institut für Normung (German standards institute)
DNS	Domain Name Service
EBCDIC	Extended Binary-Coded Decimal Interchange Code
ENV	Europäischer Normen-Vorschlag (European prestandard)
FADU	File Access Data Unit
FJAM	File Job Access Method
FSB	Forwarding Support Information Base
FSS	Forwarding Support Service
FT	File Transfer
FTAC	File Transfer Access Control
FTAM	File Transfer, Access and Management (ISO 8571)

FTPS	FTP via SSL / TLS
GPL	Gnu Public License
GSM	Global System for Mobile Communication
ISAM	Index Sequential Access Method
ISO	International Organization for Standardization
LAN	Local Area Network
LMS	Library Maintenance System
MIB	Management Information Base
MSV	Mittelschnelles Synchron Verfahren (Medium-fast synchronous method)
NDMS	Network Data Management System
NIS	Network Information Service
OSI	Open Systems Interconnection
OSS	OSI Session Service
PAM	Pluggable Authentication Modules
PEM	Privacy Enhanced Mail
PICS	Protocol Implementation Conformance Statement
PKCS	Public Key Cryptography Standards
PLAM	Primary Library Access Method
RFC1006	Request for Comments 1006
RMS	Reliant Monitor Services
SAM	Sequential Access Method
SDF	System Dialog Facility
SNA	Systems Network Architecture
SNMP	Simple Network Management Protocol
SNPA	Subnetwork Point of Attachment
SSL	Secure Socket Layer
TCP/IP	Transmission Control Protocol/Internet Protocol
TID	Transport Identification
TLS	Transport Layer Security
TNSX	Transport Name Service in Unix systems
TPI	Transport Protocol Identifier
TS	Transport System
WAN	Wide Area Network

Related publications

The manuals are available as online manuals, see *http://manuals.ts.fujitsu.com*.

openFT for Unix Systems
Managed File Transfer in the Open World
User Guide

openFT for Windows Systems
Installation and Administration
System Administrator Guide

openFT for Windows Systems
Managed File Transfer in the Open World
User Guide

openFT for Unix Systems and Windows Systems
Program Interface
Programming Manual

openFT for Unix Systems and Windows Systems
openFT-Script Interface
Programming Manual

openFT for BS2000/OSD
Managed File Transfer in the Open World
User Guide

openFT for BS2000/OSD
Installation and Administration
System Administrator Guide

openFT for BS2000/OSD
Program Interface
Programming Manual

openFT for z/OS
Managed File Transfer in the Open World
User Guide

openFT for z/OS
Installation and Administration
System Administrator Guide

CMX
Operation and Administration
User Guide

CMX
Programming Applications
Programming Manual

OSS(SINIX)
OSI Session Service
User's Guide

Index

D

P

S

T