FUJITSU

Questions and Answers
Fujitsu openFT V12.0 Software

Connectivity

X.25 configuration of BinTec Routers

- How to set up connections to partners over X.25 or X.25 over ISDN on operating systems supporting TCP/IP only?
- How to configure the TCP/IP-RFC1006 <-> X.25 conversion feature of the BinTec X1000 II/X4300 routers?

The BinTec Routers X1000 II, X1200 II, X2100, X2300/i/is/s, X2404, X2250, X4000 Series or X8500 (see http://www.teldat.de) can be used to set up a connection for openFT from any system supporting TCP/IP to a partner system over X.25 or X.25 over ISDN. TCP/IP is supported by openFT on many platforms like Windows, Linux, Solaris, HP-UX, IBM AIX, BS2000/OSD, z/OS etc. The documents which can be downloaded here describe the configuration of the BinTec X1000 II and X4300 router with the aid of an example. There are also sample configuration files (.cf) that can be loaded as a basic configuration in the corresponding router for further modifications.

X.25 configuration of FarLinX X.25 Gateways

- How to set up connections to partners over X.25 with the FarLinX X.25 Gateway?
- How to configure the TCP/IP-RFC1006 <-> X.25 conversion feature of the FarLinX X.25 Gateway?

The FarLinX X.25 Gateway (see http://www.farsite.com) can be used to set up a connection for openFT from any system supporting TCP/IP to a partner system over X.25. TCP/IP is supported by openFT on many platforms like Windows, Linux, Solaris, HP-UX, IBM AIX, BS2000/OSD, z/OS etc. The document which can be downloaded here describes the configuration of the FarLinX X.25 Gateway with the aid of an example.

FTAM protocol behavior of openFT

Application Entity Title

- The partner of openFT expects an application entity title different from the NIL APTitle (1.3.9999.1.7). How to manage it?
- openFT is expected to issue a calling APTitle different from the NIL APTitle. How to manage it?

The Application Entity Title can be used for the addressing of OSI applications. It is a parameter in ACSE (ISO 8650), consisting of APTitle, AEQualifier, AP invocation id, and AE invocation id. The invocation IDs are not used in concrete FTAM projects (as far as we know). The APTitle can be used in two formats: an object identifier, and an arbitrary (ANY) format. The optional AEQualifier can be an integer or an ANY parameter. FT only supports the object identifier resp. integer format.

Switching the Application Entity Title
The option ftmodo -ae activates/deactivates the AET (Application Entity Title).

ftmodo -ae=y	(default after installation) A "nil Application Entity Title" is included as the calling or called Application Entity Title (AET) for transfer using the FTAM protocol.
ftmodo -ae=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.

As a responder, openFT reflects the incoming called AP Title and the called AE Qualifier as responding AP Title, resp. AE Qualifier, if -ae=n is set. If ae=y is set, the responder returns the "NIL AP-Title" as responding AP Title. The incoming calling AP Title is ignored.

Specifying an Application Entity Title for the partner
An Application Entity Title other than the NIL Application Entity Title can be specified as called AET for the FTAM partner.
ftaddptn *FTAM partner name* -pa=*partner address* -id=*application entity title* (new partner)
ftmodptn *FTAM partner name* … -id=*application entity title* (existing partner)
application entity title is given in the form nn.nn.nn.nn[..nn] – a sequence of 3 to 10 integers for the Application Process Title and an optional integer for the Application Entity Qualifier. For example,
-id=1.2.250.1.999.1.1.1..5
specifies an AP Title 1.2.250.1.999.1.1.1 and an AE Qualifier 5.

Specifying an Application Entity Title for the own system
Define a REG_SZ value CALLINGAET in the registry key (for Windows platforms)
HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu Technology Solutions\openFT\CurrentVersion (>= V12)
HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\openFT\CurrentVersion (<= V11)
resp. an environment variable CALLINGAET (for UNIX platforms), and set it to your AET in the form nn.nn. ... nn (the same syntax as for the application entity title mentioned above).
This setting is valid for all commands, servers, and services started afterwards, and it will set the calling AE Title in A_ASSOCIATE request on initiator side, and the responding AE Title in A_ASSOCIATE response on responder side. As long as defined, it overrides the effect of -ae=y|n except for the called AP Title.

Recovery and Restart

- What is the difference between recovery and restart?
- Why does openFT support recovery, but not restart?

Kinds of recovery and dependencies on session functional units
ISO 8571 doesn't use the term *Class-x-recovery*. In the following, *Class-x-recovery* is meant to be the same as *recovery from Class-x-errors*. The FTAM parameter *FTAM Quality Of Service* indicates the maximum recovery facility proposed resp. negotiated for a connection.

The FTAM protocol (ISO 8571) supports three ways of recovery:
The mightiest way is the Class-III-recovery which can be applied in any case, even if one of the partner systems crash (establishing a new FTAM connection during recovery). Class-I- and Class-II-errors can be processed as Class-III-errors, thus recovery from all recoverable errors is possible.
Class-II-recovery keeps the FTAM connection, but closes and reopens the files to be transferred.
Class-I-recovery even keeps the files open, only repositioning to the last negotiated checkpoint on recovery. It is also called "restart", and it is supported in the "restart functional unit".
For class-I-recovery Minor Synchronize and Resynchronize Session functional units are required, but for the other kinds of recovery Minor Synchronize is sufficient (although the protocol handling is somewhat different if the Resynchronize FU exists, too). This is the requirement specified in the FTAM standard ISO 8571.

Transparency of Recovery Proceedings
It depends on the point of view what the question means "a temporary transport connection loss should be transparent to the application". In an enterprise file transfer like openFT, the interface of an application is like "do the transfer of the file xxxx and tell me when it is finished". In this case, all three kinds of recovery processing are invisible to the application. In the ISO 8571 protocol, the application calls an F_INITIALIZE, an F_SELECT, and so on. At this application level, n o n e of the recovery processings are completely invisible, because the user and the underlying "External File Service" must share the access to the transfer file in any way. Compared with the other kinds of recovery, the restart (Class-I) hides the greatest part of recovery proceedings.

Recovery implementations in openFT
For synchronous transfers, openFT doesn't support any recovery. For asynchronous transfers, openFT supports Class-III-Recovery, and it can be negotiated down to Class-II-Recovery. Class-I-Recovery (Restart) is not supported. The decision not to support Restart is based on the following facts:

- it cannot be applied when the FTAM connection goes down, whereas Class-III-recovery can be applied also for a simple loss of transport connection,
- the time interval between loss and re-establishing a transport connection blocks an FTAM connection (which is a precious resource in an enterprise file transfer),
- the Class-I recovery specific protocol investigations would be much larger than those of Class-II and Class-III, and there are still a lot of severe problems in the Class-I recovery protocol definitions itself.

Future File Size

- What is the file size in FTAM context?

openFT sends the parameter "future file size" in F_CREATE request when it is initiator and the sender of a file. If openFT is initiator and receiver, it sends an F_READ_ATTRIBUTE request to interrogate the "current file size" of the file to be received. This file size is used to

enable an optimal primary allocation for the receiving file (currently only for openFT mainframe implementations), and to check whether there is enough space available for this file (for openFT mainframe implementations, and for openFT version V7.0 and later).
The basis of future file size for the file to be created is the number of bytes stored in the real catalogue entry of the file to be sent from Windows resp. Unix systems. Also, if interrogated by the partner, this real catalogue entry is the basis of "current file size".
For simple binary files (ncopy/ft -b, with string significance "not significant"), this number of bytes is exactly the real number of bytes of the file. For text files with variable lengths, the interpretation of file size is not so clear. The real catalogue entry contains the total number of bytes including the line delimiters (1 per record in Unix systems, 2 per record in Windows). For GRAPHIC and GENERAL strings openFT adds escape sequences selecting the G0 and G1 character sets to each string. These escape sequences must not be added to the string length (see the definition of the Document Type simple text in FTAM part 2), thus they must not be added in the evaluation of future file size. Provided that only the bytes in the virtual file without any headers, tags, length fields and CRLF's are encountered (the most evident way to specify a virtual file size), the file length in the real catalogue can only be greater than the FTAM file size (but never less), and thus the value of future file size should be no problem.
There is one exception: the size of a GRAPHIC or VISIBLE string file containing TAB characters may grow because the TAB characters are expanded into sequences of space characters.
ISO8571/2 describes three possible behaviours of an FTAM instance when the size of a receiving file grows beyond "future file size":

- extend the file without a warning, increasing future file size
- extend the file with a warning, increasing future file size
- indicate an error, not increasing future file size

The interworking between openFT-FTAM and ICL hosts fails because of future file size when openFT is initiator and sender of a file. If future file size is not sent (for example if the sending file is stdin), the transfer to an ICL host succeeds. Therefore an option will be provided to omit future file size in F_CREATE request: in the registry key
HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu Technology Solutions\openFT\CurrentVersion (>= V12)
HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\openFT\CurrentVersion (<= V11)
a registry value NOFUFILESIZE of type REG_SZ can be specified. On openFT for Unix systems, you can specify NOFUFILESIZE as an environment variable. The specification must be done before starting the servers/services for asynchronous file transfer requests, resp. before issuing synchronous file transfer requests. NOFUFILESIZE can take the following values:

1	future file size is suppressed for all partners
0	future file size is not suppressed
prefix	future file size is suppressed for all those partners with partner names starting with prefix.

This workaround is relevant if openFT is initiator and sender of a file, and an ICL host is responder.

Workarounds for Interworking

NORTEL: Workaround for NorTel partners

- The initialization of an FTAM connection to a NorTel partner is rejected by NorTel in the FTAM layer. How to make it work?
- Why does the transfer of text files fail to NorTel partners?

The NorTel workaround in openFT-FTAM covers the following requirements specific to the NorTel implementation of the FTAM protocol:

- The ASN.1 syntax "FTAM-FADU" must be proposed in the connection establishment even if it is not needed
- The "simple binary" syntax must always be proposed
- The "simple text" syntax must not be proposed (thus text files cannot be transferred to/from NorTel switches)

On openFT for Windows, the NorTel workaround can be switched on by creating a registry value named FTAMNORTEL of type REG_SZ under
HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu Technology Solutions\openFT\CurrentVersion (>= V12)
HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\openFT\CurrentVersion (<= V11).
On openFT for Unix systems, it can be switched on by specifying an environment variable FTAMNORTEL. FTAMNORTEL can take the following values:

1	the workaround is valid for all partners
0	means that the workaround is switched off

prefix *or* 1prefix	means that the workaround is switched on for all those partners with partner names starting with prefix.
2	the workaround is valid for all partners, and the old NBS.9 document type form is used.
2prefix	the workaround is switched on for all those partners with partner names starting with prefix. For those partners, the old NBS.9 document type form is used.

This workaround is relevant if openFT is initiator and NorTel is responder.

The abstract syntax name and the document type name of NBS.9 directories exist in two variants (see note on functionality up to the V8.0 releases in 2002):

- the current form (provided in all openFT versions, and in most of the FTAM implementations)
  iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-as2(2)
  iso identified-organization oiw(14) ftamsig(5) document-type(5) file-directory(9)
- the old form (not provided by older openFT versions)
  iso identified-organization icd(9999) organization-code(1) abstract-syntax(2) nbs-as2(2)
  iso identified-organization icd(9999) organization-code(1) document-type(5) file-directory(9)

As a responder, openFT understands both NBS.9 forms provided that the initiator specifies only one form in P_CONNECT request resp. F_INITIALIZE request.

ERICSSON, CoCoNet: Minimize Access Requests

- openFT cannot receive files from some ERICSSON resp. CoCoNet partners. How to make it work?
- openFT cannot access files protected by different passwords for different access types. How to make it work?

In some cases, openFT as an initiator wants to negotiate more than one type of "access request". Some FTAM partners reject a connection proposing distinct combinations of access requests (for example READ + READ-ATTRIBUTE).

As a workaround, the access requests can be minimized by specifying an environment variable FTAMMINACQ before issuing file transfer requests. The minimizing of access requests is also very useful when files are protected by different passwords for different types of access requests, because in the openFT commands only one file or management password can be specified.
Also, if a file creation but no deletion is permitted for distinct files or users, this option may be useful in conjunction with "write-mode = NEW" (ncopy/ft -n).
FTAMMINACQ can take the following values:

1	access requests are minimized for all partners
not def.	access requests are not minimized
prefix	access requests are minimized for all those partners with partner names starting with prefix.

The access requests openFT is proposing are:

type of request	without	with minimizing
Receive file (ncopy/ft)	READ+READ-ATT	READ
Receive and delete file (ncopy/ft)	READ+DEL+READ-ATT	READ+DELETE
Send a file, NEW (ncopy/ft -n)	REPLACE(+DELETE)*	REPLACE
Send a file, OVERWRITE (ncopy/ft -o)	REPLACE+DELETE	REPLACE
Send a file, EXTEND (ncopy/ft -e)	EXTEND	EXTEND
Read file attributes (ftshw)	READ-ATTRIBUTE	READ-ATTRIBUTE
Read file directory (ftshw -d)	READ	READ
Change attributes (ftmod)	CHANGE-ATTRIBUTE	CHANGE-ATTRIBUTE
Delete file (ftdel)	DELETE	DELETE

FTAMMINACQ also affects the Permitted Actions on creating a remote file. Permitted Actions are set to:

without minimizing:	READ, REPLACE, EXTEND, ERASE, READ-ATTRIBUTE, CHANGE-ATTRIBUTE, DELETE, TRAVERSAL
with minimizing:	READ, REPLACE, EXTEND, READ-ATTRIBUTE, CHANGE-ATTRIBUTE, DELETE

Using FTAC profiles by remote partners

- By which means can a partner pass a transfer admission to openFT in order to refer to an FTAC profile?
- What to do if neither initiator identity nor filestore password can be omitted by the partner?

In order to use an FTAC profile in an openFT responder, the initiator must specify the transfer admission of this profile. He can send the transfer admission either as the initiator identity, or as the filestore password. If it is sent as initiator identity, the filestore password must be omitted, or it must be empty (string length 0). If it is sent as a filestore password, the initiator identity must be omitted.

In some cases, the initiator can omit neither the initiator identity nor the filestore password. It is possible to specify a reserved password *PROFILE in order to address FTAC profiles by the transfer admission (in the initiator identity) even if a filestore password must be given by the initiator.

ICL Hosts: Future File Size

- Interworking with an ICL host responder fails because of the future file size parameter. How can I switch off the future file size parameter request in openFT?

See ‘Future File Size in FTAM protocol behavior of openFT’.

Special termination handling for the "Elektronische Öffnung der Deutschen Bundesbank" standard

- File transfer requests are regarded as successfully completed from the client's openFT view, but not from the server's view. How to manage it?

A file transfer or file management action is successfully completed after F_CLOSE and F_DESELECT have been successfully processed. The subsequent processing is either a new transfer or file management action on the same FTAM connection, or the termination of the FTAM connection.

For this reason, FT does not care about the way of terminating the FTAM connection; once the file transfer or file management action can be regarded as successful it returns a message and a reason code indicating success. The "Elektronische Öffnung" standard, however, expects that a previously performed file transfer has to be regarded as not successful when some failure or transport disconnect occurs during the exchange of F_TERMINATE request/response.

This may lead to the situation that a transfer is regarded to be successful from the openFT initiator's view, whereas the responder's view on "Elektronische Öffnung" is that the transfer has not been performed. To overcome this situation, a new environment variable ELEKOEFF has been introduced.
ELEKOEFF can be set to one of the following values:

1	failures in the termination of the FTAM connection are always regarded as a failure of the preceding file transfer or file management action.
not def.	failures in the termination of the FTAM connection don't affect the result of the file transfer or file management.
prefix	If the specified prefix matches the partner name in the length of the prefix, failures in the FTAM connection termination will be evaluated. For all the other partners, failures in the FTAM connection termination will be ignored.

ELEKOEFF can be set as a user variable if only one user or a limited number of users interwork with a "Elektronische Öffnung" server using only synchronous request. It is better, however, to set ELEKOEFF as a system wide variable before starting the openFT servers. Windows platforms have to be restarted afterwards to make this switch work for asynchronous file transfer requests.

Warning: Don't use this ELEKOEFF option for partners which are not conformant to the "Elektronische Öffnung" standard in the sense described above, especially not for openFT responders. openFT responders don't keep any information about a file transfer request as soon as F_CLOSE and F_DESELECT response have been issued.

Special workarounds for Omikron partners

- How to achieve ASN.1 length encodings using a minimum number of octets?
- openFT cannot decode the CPA Presentation protocol element sent by Omikron
- An FTAM diagnostic code 1010 is issued by openFT. What to do?

Interworking with Omikron partners may cause a lot of problems concerning ASN.1 length encoding and FTAM protocol. Omikron expects ASN.1 length encoding using the minimum number of octets possible. On the other hand, lengths issued by Omikron are not always correct. In the FTAM protocol, Omikron returns document types and service classes which have not been proposed by the initiator in F_INITIALIZE request.

In order to overcome this situation, a new switch ASNMINLEN has been introduced. ASNMINLEN must be set as an environment variable in order to activate workarounds for these problems. If ASNMINLEN is set to any value, the ASN.1 length encoding will be minimized as expected by Omikron for any partner, and the wrong length encodings returned by Omikron will be repaired before decoding.

The workarounds in the FTAM protocol can also be activated only for specific partners. For this, ASNMINLEN can take the following values:

set ASNMINLEN=1	will activate them for all the partners
set ASNMINLEN=prefix	will activate them for all those partners with a partner name beginning with the given prefix.

It is not recommended to set ASNMINLEN if OSS is used by a performance critical transaction system (for example openUTM). Performance loss might be significant.

Local openFT behavior using FTAM

The FTAM catalogue

- Which is the background of the FTAM catalogue?
- Why are FTAM specific attributes "forgotten" by files on Windows FAT file systems?
- How to work without the FTAM catalogue?
- By which means and under which restrictions is the FTAM catalogue kept consistent with the real filestore?
- Can I specify a file password protection valid for FTAM partners?
- Can I delete a single FTAM catalogue entry without deleting the file itself?
- How can I avoid that the FTAM catalogue gets damaged?

The FTAM catalogue stores FTAM file attributes which are not known to the real system. The most important attributes stored here are format describing attributes like *contents type, universal class number* and *string format*. In the openFT interface description, those attributes are often called *file type, character set,* and *record format*. It may be more convenient to work with the FTAM catalogue, because the file format need not be specified in file transfer requests if it is stored in the FTAM catalogue entries. On the other hand, an explicit specification of a file format in a file transfer request must coincide with the information in the FTAM catalogue.
On Unix platforms, the FTAM catalogue is a subdirectory /var/openFT/<instanceName>/FTATTR containing up to 16 subdirectories with up to 16 files each. <instanceName> is normally *std*. If an FTAM file is created, also an FTAM catalogue entry is created in this database. If the file is deleted by means of FTAM, also the corresponding catalogue entry is deleted. If the file is deleted locally, the corresponding catalogue entry stays alive. These catalogue entries without a real file may cause the following problems: first, the FTAM catalogue may grow and grow by the time, and second, a locally created file with a name earlier used may "inherit" improper attributes. The process ftvfsm helps to avoid those problems as far as possible. It is started together with the openFT servers, and it erases the catalogue entries without an existing file once a day. In some applications, the cleaning once a day seems to be not enough. ftvfsm may

also be called as a command, and it could be started (for example) once an hour by an automatism like crontab. It is also possible to work without the FTAM catalogue if you erase the directory */var/openFT/<instanceName>/FTATTR*.
On Windows, each FTAM catalogue entry is an "appendix" to an NTFS file. The problem of keeping the FTAM catalogue consistent with the existing files doesn't exist here. Non-NTFS files cannot have an FTAM catalogue entry. It is also possible to switch off the FTAM catalogue function on Windows platforms by specifying a value BYPASSVFS of type REG_SZ in the registry key
HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu Technology Solutions\openFT\CurrentVersion (>= V12)
HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\openFT\CurrentVersion (<= V11)
set to "1".
For NTFS files on Windows and for files on Unix platforms, it is possible to see the contents of an FTAM catalogue entry using *ftshwf -l <filename>*. If and only if there is no FTAM catalogue entry, the output will contain neither a *BINARY-FILE* nor a *CHARACTERSET* nor a *RECORD-FORMAT* specification. In order to create or modify an FTAM catalogue entry the command *ftmodf* can be used. If no file format attributes are specified in this command, they are set to *CHARACTERSET=g RECORD-FORMAT=v* if the catalogue entry didn't exist before, else they remain unchanged.
You can specify a file access password using the parameter *-np=<password>* in the *ftmodf* command. This password is significant only for file transfer requests (sending and receiving) and deletion requests initiated by remote partners. *ftmodf -np=@n* removes this password protection.
It is also possible to delete FTAM catalogue entries without deleting the associated file: *ftmodf <filename> -np=@d* will do this.
FTAM catalogue entries are not set by FT protocol requests, but they may affect them. For example, a file created as a binary file by an FTAM protocol request can be transferred by an FT protocol request only if the binary option is set.

Tabulator expansion and Escape Sequences in FTAM

- On transferring text files, tabulators are replaced by sequences of blanks. How to avoid this?
- Why are escape sequences added in graphic resp. general string text files for transfer?
- How are "illegal characters" (in the sense of the character set) processed?

FTAM text files may contain control characters, and they may contain escape sequences, for example for switching between character sets like ISO8859-1, ISO8859-2. It depends on the *universal class number* resp. the *character set* whether control characters and/or escape sequences are allowed or not.
In principle, openFT doesn't check these rules, that means, neither a local file to be transferred nor a data stream is checked for "illegal characters". There are two facts, however, which may modify the binary contents of a file.

The Tabulator Expansion
Files of the character set types *graphic string* and *visible string* cannot contain control characters; especially they cannot contain tab signs. Each tab sign is substituted therefore by one to eight space characters, as if there were tabulator positions 1,9,17 and so on. This behavior is compatible with the tabulator expansion in the FT protocol solution. *IA5 string* and *general string* files leave tab signs unchanged.

Escape Sequences
In *graphic string* and *general string* files it is possible to specify character set switching escape sequences. These escape sequences are not part of the text, but they may switch proper character imaging devices from one representation of characters (like West European) to another one (like Greek). openFT-FTAM as a sender of such a file assumes that there is an ISO 8859-1 file to be sent, and the appropriate escape sequences are added as a header of each string. openFT-FTAM as a receiver removes those escape sequences which indicate ISO8859-1 or somewhat compatible. More exactly, the following escape sequences are removed:

- 1B 28 42
- 1B 28 40
- 1B 2D 41
- 1B 2D 7E
- 1B 21 40
- 1B 21 7E
- 1B 7E
- (since V10:) 1B 25 47

Any other escape sequence is preserved. In *visible string* and *IA5 string* files, there are no escape sequences, and openFT-FTAM leaves any sequence of characters starting with ESC (1B) unchanged.
The insertion and removal of escape sequences can be switched off even for *graphic string* and *general string*. Setting the environment variable NOESCSEQ to 3 will do this. NOESCSEQ=1 switches off the insertion, but does not affect the removal of escape sequences. NOESCSEQ=2 keeps all the escape sequences on file receiving, but does not affect the insertion of escape sequences. It is recommended to set NOESCSEQ as a system wide environment variable if needed.

Conclusion
"Legal" characters are always mapped correctly. The maximum transparency concerning the binary representation of a text file including "illegal characters" can be achieved using the *IA5 string*: neither the tabulator expansion nor the processing of escape sequences is effective in this case.

Ambiguous document type specification

- Receiving files from some Alcatel switches does not work

Some Alcatel switches return document type specifications of the type "simple binary", but including a universal class number. This is ambiguous, because a universal class number makes no sense for a binary file.
openFT up to V11.0B00 always rejects this document type. Starting from openFT V11.0B10 and V12.0A00, openFT accepts this document type if the parameter "text" or "binary" is explicitly specified in the file transfer request.

Activity Identifier in requests without recovery functional unit

Starting from openFT V12.0, an activity identifier is also specified if no recovery has been negotiated for a request - in order to improve diagnostic capabilities in network problems. This is not really in the scope of ISO8571, but usually it causes no problems.
If some problems occur, this feature can be switched off by specifying an environment variable (Unix) resp. a registry value (Windows) in HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu Technology Solutions\openFT\CurrentVersion: FTAMSYNCNOACTID=1.

Hidden Functions

Hidden configuration parameters in the ftmodo command

- How can I modify the timeout for idle connections?
- How can I modify the time intervals between checkpoints?

ftmodo.....	min. value	default value	max. value	explanation
-hf				must be given in conjunction with every hidden parameter.
-to=	1	10	1500	(Timeout in minutes) If a connection stays idle for more than this number of minutes, it will be aborted.
-ci=	2	20	120	(Checkpoint Interval) the time in seconds between checkpoints.

The settings of the hidden parameters of the ftmodo command can be seen by ftshwo –csv (in the fields ConTimeout resp. ChkpTime).

Hidden parameters in the ncopy/ft command

- On transferring text files, tabulators are replaced by sequences of blanks. How to avoid this if there is no FTAM catalogue?
- ZKA data exchange requires special file transfer formats. How to manage it?

For FTAM partners, you can specify an option *-g* as an alternative to *-t, -b or -u* in an ncopy or ft command. This option claims the following file format attribute combination for the transfer:

- document type: simple text
- universal class number: General
- string significance: not significant

This option can be used to avoid tabulator expansion when a file is to be sent and when there is no FTAM catalogue entry for this file possible. If an FTAM catalogue entry can be assigned to the file, it is better to set it by means of the command
ftmodf <filename> -ft=t -cs=c -rf=u
before sending the file instead of using the -g option.
It may be required to avoid the escape sequences for character set switching. This can be done by the selection of the universal class number IA5 String or Visible String. Additionally to the setting of the universal class number in the FTAM catalogue, there exists a more convenient non-official way starting from openFT version V8.0B for Unix systems and Windows:
-cs=c | -cs=i | -cs=v
in the ncopy resp. ft command sets the universal class number for the transfer to General resp. IA5 resp. Visible String. This switch must be used in the following combinations:

- *-t -cs=v* for Visible string with variable string significance
- *-g -cs=i* for IA5 String with not significant
- *-g -cs=c* for General String with not significant (same as -g)
- *-t -cs=i* for IA5 String with variable
- *-t -cs=c* for General String with variable

If the ncopy/ft parameter *-r=fnnn* is combined (nnn describing a string length) with -t, the string significance is fixed instead of variable. The combinations of IA5 or General String with variable or fixed string significance are new transfer formats (not supported before version V8.0B), but in practice they are very similar to the transfer format of IA5 or General String with Not Significant: <CRLF> are issued resp. interpreted as line delimiters. The file is sent in string portions of maximum string length (if given); on receiving these string portions have no significance for the file structure. In the FTAM catalogue, the files are created with not significant (independent of the transfer format's string significance). On the other hand, files with "not significant" given in the FTAM catalogue, or without an FTAM catalogue entry can be sent in variable or fixed transfer format (with General or IA5 string) by setting these parameter combinations in the ncopy resp. ft command or inbound in the F_OPEN_request.
Warning: Don't use the -g option or the -cs= switch for NEA transfer requests. Results will be unpredictable.

Hidden parameters in the program interface for transfer requests

- ZKA data exchange requires special file transfer formats. How to manage it?
- ZKA data exchange requires the output of FTAM diagnostic further details. How to manage it?

It is also possible to set some of the transfer formats in the C program interface of openFT. In the structure *ft_transpar, filetype* can take one of the following values:

filetype	like ... in the ncopy command
FT_NOTYPE	(no file type given)
FT_TEXT	-t
FT_BINARY	-b
FT_USER	-u
'v'	-t -cs=v
'i'	-g -cs=i
'c'	-g

The fixed string significance format can be selected by ORing the value 0x00100000 to the string length given in *maxrecsize*.
The variable string significance format can be selected by ORing the value 0x00200000 to the string length given in *maxrecsize*.

Warning: Don't use values other than FT_NOTYPE, FT_TEXT, FT_BINARY, or FT_USER for *filetype* in *ft_transpar* for NEA transfer requests. Results will be unpredictable.

Encryption

Key Management

- Have keys to be exchanged for encryption even if no partner authentication is required?
- For encryption only, is it necessary to store the public key in the *syskey* directory?

The instance which is the initiator sends its public RSA key to the partner. The partner creates an AES key, encrypts it with the public RSA key of the initiator and sends it back to the initiator. The initiator can decrypt the encrypted AES key with his private RSA key.
Then the request description data is transferred encrypted. If the user data should be transferred encrypted, too (option -c is set), it will be encrypted with the AES key, provided that openFT-CR is installed.
So, to encrypt the request description data without partner authentication, the keys do not have to be exchanged by the openFT administrator explicitly.

It is sufficient to generate an RSA key by the command ftcrek (CREATE-FT-KEY-SET). The generated RSA keys need not be stored in the library/directory *syskey*. This library/directory is used for authentication only.

Is the identification of a partner necessary to enable encryption?
The identification is not relevant either. It is necessary for authentication only.

Encryption Status

- Is the identification of a partner necessary to enable encryption?
- Is encryption automatically enabled for all partners if a pair of keys exists?
- Is it recognizable if encryption is enabled or not?
- If a local key pair set is deleted, what will happen to the encryption?

If a pair of keys is available on both partners, the request description data is encrypted automatically, else no encryption of the request description data will be done.
If partner authentication is wanted, the public keys have to be copied to the *syskey* directory. Then it is possible to control via the command ftmodp (MODIFY-FT-PARTNER) which partner must use authentication.
Generally, FTAC (openFT-AC) has to be used, if either encryption or authentication is to be enforced. With the definition of openFT profiles the encryption or authentication can be forced by the administrator.

Whether or not encryption is used can be seen in the logging of „data transfer requests and authorization checks“ in the long output form under SEC-OPTS.

openFT supports a maximum of three key pair sets at a time. If at least one key pair set is left after the deletion, encryption will be unharmed. If the last local key pair set is deleted, however, all requests will run unencrypted, i.e. neither the request data nor the file contents will be encrypted. So be careful when deleting key pair sets!

Windows

Problems with syntactically correct openFT commands

- Sometimes openFT commands behave differently in a Windows batch file

A syntactically correct openFT command containing a ‘!’ can be executed correctly in a command window. But the same command in a batch file produces the error message “parameter is missing“.

The reason is the delayed variable expansion. If delayed variable expansion is activated, ‘!’ has a special meaning for the evaluation of environment variables, i.e. the character string after the exclamation mark up to the next exclamation mark will be interpreted as a variable name. Normally this string will not be a valid variable name; therefore the exclamation mark will simply be removed, and the openFT command will end with a syntax error.

To avoid this problem the exclamation mark has to be preceded by the escape character ‘^’ (circumflex accent / spacing circumflex). Additionally the parameter has to be imbedded in double quotes.

Example:
ncopy local.txt "partner^!remote.txt" transferadmission

ZIP Archives

Examples for openFT (Windows) V12.0 and BS2000/OSD

Sample 1: ZIP archive t1.zip generated in Windows should be transferred to BS2000 system host1. The file name should be kept under user id that belongs to the admission profile with the transfer admission SendToBS2. For sending the file use the parameter –tff=b which is available as of openFT V11. In BS2000 all files with the suffix .txt should be extracted.

```
WIN:    ncopy t1.zip host1!% SendToBS2 -tff=b
BS2:    /START-ZIP-MANAGER
        //OPEN-ZIP-CONTAINER CONTAINER=t1.zip,MODE=*UPDATE,FORMAT=*WINZIP
        //SHOW-FI-ATTRIB
        //EXTRACT-FILE FILE-NAME='*.txt',DATA-TYPE=*CHAR   #Case sensitivity!
        //END
        /SHOW-FILE fix.txt
```

Sample 2: The ZIP archive t2.zip should be created in BS2000 and files fix.txt and file.test should be added. This archive will be transferred to Windows system winpc and stored there under the transfer admission SendToWin. In the Windows system the archive can be processed by winzip.

```
BS2:    /START-ZIP-MANAGER
        //OPEN-ZIP-CONT CONT=t2.zip,MODE=*UPDATE(STATE=*NEW),FORMAT=*WINZIP
        //ADD-FILE fix.txt
        //ADD-FILE file.test
        //end
        /FTSCOPY TO,winpc,(t2.zip),*A('t2.zip',,'SendToWin'),TARGET-FILE-FORMAT=*BLOCK
WIN:    double click to ZIP archive and extract
```

Create ZIP archives in BS2000/OSD

In BS2000/OSD you have to use the command /START-ZIP-MANAGER. To create ZIP archives use //OPEN-ZIP-CONTAINER and to add files use //ADD-FILE. With //SHOW-FILE-ATTRIB you can retrieve the properties of the contained files. ZIP archives to be accessed in Windows have to be created in BS2000 with parameter FORMAT *WINZIP.

Sample: Create a new ZIP archive

```
/START-ZIP-MANAGER
//OPEN-ZIP-CONTAINER CONTAINER=bs2.zip,MODE=*UPDATE(STATE=*NEW),FORMAT=*WINZIP
//ADD-FILE chs.
//SHOW-FILE-ATT
```

Sample: Open an existing ZIP archive

```
/START-ZIP-MANAGER
//OPEN-ZIP-CONTAINER CONTAINER=bs2.zip,MODE=*UPDATE
//SHOW-FILE-ATT
```

Sample: Extract the file fix1.txt out of a ZIP-archive

```
//OPEN-ZIP-CONTAINER CONTAINER=bs2.zip,MODE=*UPDATE
//SHOW-FILE-ATT
//EXTRACT-FILE FILE-NAME='fix.txt',DATA-TYPE=*CHAR    # Case sensitivity!
```

Transferring ZIP archives

Creation of ZIP archive in BS2000 system

1. Initiator is BS2000: Send to Windows openFT V10 without parameter: Open/Change the file by winzip
2. Initiator is BS2000: Restore from Windows system with parameter
   TARGET-FILE-FORMAT=*BLOCK, file can be opened and changed with BS2ZIP.
   Receiving the file from openFT V10 without that switch, the file can't be opened with BS2ZIP, because of SAM format
3. Initiator is BS2000: Send file to Windows openFT as of V11 with parameter
   TARGET-FILE-FORMAT=*BLOCK, file is processable with winzip
4. Initiator is BS2000: Restore from Windows openFT as of V11 with parameter
   TARGET-FILE-FORMAT=*BLOCK, file is processable with BS2ZIP

Creation of ZIP archive in Windows system

1. Initiator is Windows: Send the file to BS2000 with parameter -tff=b,
   sending without that parameter doesn't work because a SAM file would be created in BS2000 otherwise.
2. Initiator is BS2000: Receive the file with parameter TARGET-FILE-FORMAT=*BLOCK