Deutsch

FUJITSU Software

# openFT V12.1

## Konzepte und Funktionen

Benutzerhandbuch

Ausgabe Juli 2017

## Kritik… Anregungen… Korrekturen…

Die Redaktion ist interessiert an Ihren Kommentaren zu diesem Handbuch. Ihre Rückmeldungen helfen uns, die Dokumentation zu optimieren und auf Ihre Wünsche und Bedürfnisse abzustimmen.

Sie können uns Ihre Kommentare per E-Mail an manuals@ts.fujitsu.com senden.

## Zertifizierte Dokumentation nach DIN EN ISO 9001:2008

Um eine gleichbleibend hohe Qualität und Anwenderfreundlichkeit zu gewährleisten, wurde diese Dokumentation nach den Vorgaben eines Qualitätsmanagementsystems erstellt, welches die Forderungen der DIN EN ISO 9001:2008 erfüllt.

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

## Copyright und Handelsmarken

# Inhalt

# 1 Einleitung

Die openFT-Produktfamilie überträgt und verwaltet Daten

– automatisiert
– sicher
– kostengünstig.

Das sichere und komfortable Übertragen von Daten - der File Transfer - ist eine wichtige Funktion in einem leistungsfähigen Rechnerverbund. Innerhalb eines Unternehmens sind die Arbeitsplatz-PCs untereinander vernetzt und meistens mit einem Mainframe, einem Unix-basierten Server oder einem Windows-Server gekoppelt. So kann ein großer Teil der Verarbeitungsleistung direkt am Arbeitsplatz erbracht werden, während für andere Fälle Daten via File Transfer zum Mainframe übertragen und dort weiterverarbeitet werden. Dabei können die Standorte der einzelnen Rechner weit voneinander entfernt liegen. Fujitsu bietet eine umfangreiche File-Transfer-Produktlinie, die openFT-Produktfamilie, für folgende Systemplattformen:

- BS2000®
- Linux® (x86 und x86_64 / IBM z Systems), Solaris™ (SPARC®/Intel™), AIX®, HP-UX®
- Microsoft® Windows™ 8.1, 10, Windows Server 2012 R2, Windows Server 2016
- z/OS (IBM®)

## 1.1 Kurzbeschreibung des Produkts

Die openFT-Produktfamilie besteht aus folgende Produkten:

**FUJITSU Software openFT (Unix-Systeme)** ist das File-Transfer-Produkt für Rechner mit einem Unix-basierten Betriebssystem.

**FUJITSU Software openFT (Windows)** ist das File-Transfer-Produkt für Rechner mit den Betriebssystemen Windows von Microsoft.

**FUJITSU Software openFT (BS2000)** ist das File-Transfer-Produkt für Rechner mit dem Betriebssystem BS2000.

**FUJITSU Software openFT (z/OS**) ist das File-Transfer-Produkt für Rechner mit dem Betriebssystem z/OS.

Alle openFT-Produkte kommunizieren untereinander über das durch Fujitsu festgelegte openFT-Protokoll (früher nur als FTNEA-Protokoll bekannt). Da auch etliche andere FT-Produkte dieses Protokoll unterstützen, bestehen vielfältige Kopplungsmöglichkeiten zu anderen Betriebssystemen.

openFT lässt auf folgende Transportprotokolle zu:

- TCP/IP
- ISO TP0/2 (nicht auf z/OS)
- ISO TP4 (nicht auf z/OS)
- SNA (nur auf z/OS)

Der Funktionsumfang von openFT kann erweitert werden durch:

- FTAC:

  FTAC bietet einen erweiterten Zugangs- und Zugriffsschutz. FTAC steht für **F**ile **T**ransfer **A**ccess **C**ontrol.
  FTAC wird auf BS2000-Systemen und auf z/OS durch das Zusatzprodukt openFT-AC realisiert.

- openFT-FTAM (nicht auf z/OS verfügbar):

  openFT-FTAM unterstützt das in der ISO-Norm FTAM (File Transfer Access and Management) standardisierte File-Transfer-Protokoll. Dadurch sind weitere Kopplungen zu Systemen anderer Hersteller möglich, deren File-Transfer-Produkte diese Norm ebenfalls unterstützen.

- openFT-FTP:

  openFT-FTP unterstützt die FTP-Funktionalität. Damit ist eine Kopplung zu beliebigen FTP-Servern möglich.

## 1.2 Zielsetzung und Zielgruppen des Handbuchs

Dieses Handbuch dient dazu, die Produktfamilie openFT kennen zu lernen und in die konkrete Arbeit mit openFT einzusteigen. Es richtet sich insbesondere an diejenigen, die mit openFT noch nicht vertraut sind. Aber auch wenn Sie openFT bereits kennen und einsetzen, können Sie dieses Handbuch verwenden, um sich einen Überblick über die Funktionsbreite und Leistungsfähigkeit des Produkts zu verschaffen.

In diesem Handbuch stehen nicht die syntaktischen Feinheiten einzelner Anweisungen oder die Details spezifischer Schnittstellen im Vordergrund. Vielmehr soll ein Überblick über die Leistungsfähigkeit und die Einsatzmöglichkeiten von openFT gegeben werden. Mit diesem Überblick ausgestattet, werden Sie sich in den übrigen Handbüchern der openFT-Reihe leicht zurechtfinden.

## 1.3 Handbuchkonzept von openFT

**openFT - Konzepte und Funktionen**

Dieses Handbuch richtet sich an alle, die den Leistungsumfang von openFT kennen lernen und die Funktionsweise verstehen möchten. Es beschreibt:

– das Konzept von openFT als Managed File Transfer
– den Leistungsumfang und die grundsätzlichen Funktionen der openFT-Produktfamilie
– die openFT-spezifischen Fachwörter

**openFT (Unix- und Windows-Systeme) - Installation und Betrieb**

Dieses Handbuch richtet sich an FT-, FTAC- und ADM-Verwalter auf Unix- und Windows-Systemen. Es beschreibt:

– die Installation von openFT und seinen optionalen Komponenten
– Betrieb, Steuerung und Überwachung des FT-Systems und der FTAC-Umgebung
– die Konfiguration und den Betrieb eines Fernadministrations-Servers und eines ADM-Trap-Servers
– wichtige CMX-Kommandos auf Unix-Systemen

**openFT (BS2000) - Installation und Betrieb**

Dieses Handbuch richtet sich an FT- und FTAC-Verwalter auf BS2000-Systemen. Es beschreibt:

– die Installation von openFT und seinen optionalen Komponenten auf dem BS2000-System.
– Betrieb, Steuerung und Überwachung des FT-Systems und der FTAC-Umgebung
– die Abrechnungssätze

**openFT (z/OS) - Installation und Betrieb**

Dieses Handbuch richtet sich an FT- und FTAC-Verwalter auf dem z/OS. Es beschreibt:

– die Installation von openFT und seinen optionalen Komponenten, einschließlich der notwendigen Voraussetzungen für den Einsatz des Produkts.
– Betrieb, Steuerung und Überwachung des FT-Systems und der FTAC-Umgebung
– die Meldungen von openFT und openFT-AC für den FT-Verwalter
– weitere Informationsmöglichkeiten für den FT-Verwalter, z.B. die Abrechnungssätze und die Logging-Information

**openFT (Unix- und Windows-Systeme) - Kommandoschnittstelle**

Dieses Handbuch richtet sich an den openFT-Benutzer auf Unix- und Windows-Systemen und beschreibt:

– die Konventionen für den File Transfer zu Rechnern mit verschiedenen Betriebssystemen
– die openFT-Kommandos auf Unix- und Windows-Systemen
– Meldungen der verschiedenen Komponenten

Die Beschreibung der openFT-Kommandos gilt auch für die POSIX-Schnittstelle auf BS2000-Systemen.

**openFT (BS2000) - Kommandoschnittstelle**

Dieses Handbuch richtet sich an den openFT-Benutzer auf BS2000-Systemen und beschreibt:

– die Konventionen für den File Transfer zu Rechnern mit verschiedenen Betriebssystemen
– die openFT-Kommandos auf BS2000-Systemen
– Meldungen der verschiedenen Komponenten

**openFT (z/OS) - Kommandoschnittstelle**

Dieses Handbuch richtet sich an openFT-Benutzer auf z/OS-Systemen und beschreibt:

– die Konventionen für den File Transfer zu Rechnern mit verschiedenen Betriebssystemen
– die openFT-Kommandos auf z/OS
– die Menüschnittstelle für den FT-Verwalter und den FT-Benutzer
– die Programmschnittstelle für den FT-Benutzer
– Meldungen der verschiedenen Komponenten

**openFT (BS2000) - Programmschnittstelle**

Dieses Handbuch richtet sich an den openFT-Programmierer und beschreibt die openFT- und openFT-AC-Programmschnittstellen auf BS2000-Systemen.

**openFT (Unix- und Windows-Systeme) - C- und Java-Programmschnittstelle**

Dieses Handbuch richtet sich an C- und Java-Programmierer auf Unix- und Windows-Systemen. Es beschreibt die C-Programmschnittstelle sowie die Grundzüge der Java-Schnittstelle.

**openFT (Unix- und Windows-Systeme) - openFT-Script-Schnittstelle**

Dieses Handbuch richtet sich an XML-Programmierer und beschreibt die XML-Anweisungen der openFT-Script-Schnittstelle.

Viele der in den Handbüchern beschriebenen Funktionen können auch über die grafische Oberfläche von openFT, dem openFT Explorer, ausgeführt werden. Der openFT Explorer steht auf Unix- und Windows-Systemen zur Verfügung. Damit können Sie unabhängig vom lokalen System den Betrieb und die FTAC-Umgebung ferner openFT-Installationen auf beliebigen Plattformen bedienen, steuern und überwachen. Mit dem openFT Explorer wird eine ausführliche Online-Hilfe ausgeliefert, in der die Bedienung mit allen Dialogen beschrieben wird.

## 1.4 Änderungen gegenüber der vorigen Version

In diesem Abschnitt finden Sie die Änderungen von openFT V12.1 gegenüber openFT V12.0A.

Die funktionalen Erweiterungen der openFT-Kommandos, die entweder den Verwalter oder den Benutzer betreffen, stehen auch im openFT Explorer zur Verfügung, der auf Unix- und Windows-Systemen ausgeliefert wird. Details finden Sie in der zugehörigen Online-Hilfe unter dem Thema *Neue Funktionen*.

Auf z/OS stehen die funktionalen Erweiterungen auch im Menüsystem (Panels) zur Verfügung.

### 1.4.1 Änderungen für alle Plattformen

- Erweiterte Unicode-Unterstützung

  Auf allen Unicode-fähigen Systemen dürfen Dateinamen, FTAC-Zugangsberechtigungen und Folgeverarbeitungen auch Unicode-Zeichen enthalten. Dazu wurde die neue Funktion „Codierungsmodus“ eingeführt, um die Unicode-Namen auf den beteiligten Systemen korrekt darzustellen.

  Die Kommandoschnittstellen wurden wie folgt erweitert:

  – Alle Plattformen: Das neue Feld FNC-MODE bei der Langausgabe von Logging-Sätzen zeigt den Codierungsmodus für den Dateinamen an (Kommandos *ftshwl*, SHOW-FT-LOGGING-RECORDS und FTSHWLOG). Auf BS2000-Systemen wurden die OPS-Variablen um die neuen Elemente FNC-MODE und FNCCS erweitert.

  – Unix- und Windows-Systeme:
    – Neue Option *-fnc*, um den Codierungsmodus bei einem Dateiübertragungs-, Dateimanagement oder Administrationsauftrag festzulegen. Diese Option steht bei den Kommandos *ft*, *ftadm*, *ftcredir*, *ftdel*, *ftdeldir*, *ftexec*, *ftmod*, *ftmoddir*, *ftshw* und *nopy* zur Verfügung.
      Der Codierungsmodus wird (zusätzlich zu *ftshwl*) bei folgenden Kommandos angezeigt: *ftshw* und *ftshwr* (Feld FNC-MODE)
      Mit *ftshw -sif* wird die Anzahl der nicht abbildbaren Dateinamen angezeigt.
    – Neues Attribut *CmdMode* in der Konfigurationsdatei eines Fernadministrations-Servers um den (empfohlenen) Codierungsmodus für administrierte openFT-Instanzen festzulegen. Der Codierungsmodus wird beim Kommando *ftshwc* im Feld MODE angezeigt.
      Diese Funktion steht auch im Konfigurations-Editor des openFT Explorers zur Verfügung.

  - Auf Unix-Systemen lässt sich für Inbound-Aufträge zusätzlich der Zeichensatz einstellen, der im Zeichenmodus verwendet werden soll. Dazu wurde die neue Option *-fnccs* im Kommando *ftmodo* eingeführt.
    Bei *ftshwo* wird im Feld FN-CCS-NAME der Zeichensatz angezeigt, der aktuell für Inbound-Aufträge im Zeichenmodus eingestellt ist.

- Bei Inbound-Aufträgen wird in der Langausgabe und der CSV-Ausgabe der Logging-Sätze im neuen Feld PTNR-ADDR die Adresse des Partnersystems angezeigt. Auf BS2000-Systemen wird die Partneradresse auch in der OPS-Variable PARTNER-ADDRESS angezeigt

- Deaktivierung der Wiederanlauf-Funktion

  Für asynchrone Dateiübertragungs-Aufträge über das openFT- und FTAM-Protokoll kann die Wiederanlauf-Funktion deaktiviert werden. Der Wiederanlauf kann partnerspezifisch für Outbound-Aufträge und global für Inbound- und Outbound-Aufträge eingestellt werden. Dazu wurden folgende Kommandos geändert:

  Unix- und Windows-Systeme:
  – *ftaddptn* und *ftmodptn*: Neue Option *-rco*
  – *ftmodo*: Neue Optionen *-rco* und *-rci*

  BS2000- und z/OS-Systeme:
  – ADD-/MODIFY-FT-PARTNER bzw. FTADDPTN/FTMODPTN:
    Neuer Operand RECOVERY-OUTBOUND
  – MODIFY-FT-OPTIONS bzw. FTMODOPT:
    Neue Operanden RECOVERY-OUTBOUND und RECOVERY-INBOUND

- RSA-Mindestschlüssellänge für das openFT Protokoll

  Eine openFT Instanz kann eine RSA-Mindestschlüssellänge zur Verschlüsselung der openFT-Session verlangen. Die Mindestschlüssellänge kann in den Betriebsparametern festgelegt werden. Dazu wurden folgende Kommandos geändert:

  Unix- und Windows-Systeme:
  – *ftmodo*: Neue Option *-klmin*

  BS2000- und z/OS-Systeme:
  – MODIFY-FT-OPTIONS bzw. FTMODOPT: Neue Parameter RSA-PROPOSED und RSA-MINIMUM beim Operanden KEY-LENGTH

- AES-Mindestschlüssellänge für das openFT Protokoll

  Eine openFT Instanz kann eine AES-Mindestschlüssellänge zur Verschlüsselung der openFT-Session verlangen. Die Mindestschlüssellänge kann in den Betriebsparametern festgelegt werden.

Dazu wurden folgende Kommandos geändert:

Unix- und Windows-Systeme:
– *ftmodo*: Neue Option *-aesmin*

BS2000- und z/OS-Systeme:
– MODIFY-FT-OPTIONS bzw. FTMODOPT: Neuer Parameter AES-MINIMUM beim Operanden KEY-LENGTH

### 1.4.2 Änderungen für Unix- und Windows-Plattformen

- Übertragung von Dateiverzeichnissen:
  – Zwischen Unix- und Windows-Systemen können Dateiverzeichnisse übertragen werden. Dazu wurden die Kommandos *ft* und *ncopy* um die Option *-d* erweitert.
  – Das neue Feld PROGRESS in der Ausgabe von *ftshwr* zeigt den aktuellen Bearbeitungsstand bei der Übertragung von Verzeichnissen an.
  – Die neue Option *ftmodo -ltd* wurde eingeführt, um den Umfang des Logging bei der Verzeichnisübertragung einzustellen.
  – Der neue Wert *ftshwl -ff=T* selektiert Logging-Sätze von Aufträgen zur Verzeichnisübertragung. Zusätzlich wurde die Ausgabe von *ftshwl* erweitert um das Feld TRANSFILE (Langausgabe) sowie um die FT-Funktionen TD, SD, SF (Kurzausgabe) bzw. um den Wert FUNCTION=TRANSFER-DIR (Langausgabe).
- Übertragung mehrerer Dateien mit FTAM:

  Zwischen Unix- und Windows-Systemen können über das FTAM-Protokoll mehrere Dateien synchron übertragen werden. Dies wird über eine spezielle Dateinamen-Syntax des Kommandos *ncopy* gesteuert.
- Erweiterung der openFT-Script-Kommandos
  – Der FT-Verwalter kann Grenzwerte für openFT-Script-Aufträge einstellen. Dazu wurde das Kommando *ftmodsuo* um die Optionen *-u*, *-thl* und *-ftl* erweitert.
  – Die aktuell eingestellten Grenzwerte werden bei *ftshwsuo* ausgegeben.
- Das Kommando *ftshwk* zeigt für öffentliche Schlüssel von Partnersystemen den Partnernamen an.
- Unterstützung von FarSync X.25

  Auf Linux und Windows werden FarSync X.25-Karten vom Hersteller FarSite direkt durch openFT unterstützt. PCMX wird dafür nicht mehr benötigt.
  Zusätzlich wird auf Linux durch den Einsatz der FarSync XOT-Runtime die Verbindungsmethode XOT (X.25 über TCP/IP) unterstützt.
  Dazu wurden die Kommandos *ftaddptn*, *ftmodptn*, *ftmodo*, *ftshwptn* und *ftshwo* erweitert.

- Erweiterte Unterstützung des Application Entity Title

  Der Application Entity Title (AET) kann jetzt auch für FTAM-Partner zur Absenderüberprüfung verwendet werden. Dazu wurde das Kommando *ftmodo* geändert, indem die Partner-Check-Option *-ptc* erweitert sowie die Option *-aet* zur Spezifizierung des AET hinzugefügt wurde. Das Kommando *ftshwo* wurde um die Option *-ae* erweitert.

- Sonstige Änderungen

  – Geänderte Absenderüberprüfung für Partner, die über IPV6 mit Scope-ID oder X.25 mit Anschlussnummer adressiert werden. Damit ist immer eine eindeutige Identifizierung über die Partneradresse möglich.

  – Das Kommando *ft_mget* wurde um die Option *-case* erweitert, welche die Berücksichtigung der Groß- und Kleinschreibung in der Dateinamenstruktur steuert.

  – Der ADM-Verwalter kann seine Fernadministrations-Berechtigung jetzt auch zurückgeben (Kommando *ftmoda -admpriv=n)*. Die Konfiguration des Fernadministrations-Servers bleibt erhalten.

### 1.4.3 Änderungen für Unix-Plattformen

- Einbenutzerbetrieb

  Auf Unix-Systemen kann der Administrator mit dem neuen Kommando *ftsetmode* vom standardmäßigen Mehrbenutzerbetrieb in den Einbenutzerbetrieb umschalten und umgekehrt. Im Einbenutzerbetrieb läuft openFT vollständig unter einer bestimmten Kennung, der sogenannten openFT-Kennung, die zugleich FT- und FTAC-Verwalter ist. Zum Anlegen und Verwalten weiterer openFT-Instanzen im Einbenutzerbetrieb wurden die Kommandos *ftcrei* und *ftmodi* um die Option *-ua* zur Angabe der Benutzerkennung einer neuen Instanz erweitert.

- openFT-Freigabe für 64-Bit-Linux

- SNMP wird auf Unix-Plattformen nicht mehr unterstützt.

### 1.4.4 Änderungen für BS2000-Systeme und zOS

- Neues Kommando GET-REMOTE-FILES (BS2000-Systeme) bzw. FTMGET (z/OS), um mehrere mit Wildcards spezifizierte Dateien synchron oder asynchron von einem fernen System zu holen.

- Neues Diagnose-Kommando FTPING auf BS2000-POSIX und z/OS zum Testen einer openFT-Verbindung zu einem fernen Partner.

### 1.4.5 Änderungen für zOS

- Das PARM-Element der z/OS Parameterdatei wurde wie folgt geändert:
  - Neues Schlüsselwort JOB_JOBCLASS für Folgeverarbeitungs-Jobs, Vorverarbeitungs- und Nachverarbeitungs-Jobs sowie Print-Jobs.
  - Neues Schlüsselwort LISTPARM für die Einstellung eines Standard-Druckers (LISTING=*STD in einem FT-Auftrag).
  - Das Schlüsselwort JOB_MSGCLASS gilt jetzt auch für Vorverarbeitungs- und Nachverarbeitungs-Jobs.
- In FTBATCH kann ab z/OS V2.1 der PARMDD-Parameter anstelle des Parameters PARM verwendet werden.
- NCOPY und FTACOPY: neuer Operandenwert LISTING=*STD bei LOCAL-PARAMETER, um den mit LISTPARM definierten Drucker zuzuordnen.
- openFT (z/OS) unterstützt jetzt Hostnamen mit einer Länge von bis zu 80 Zeichen. Dies gilt sowohl für die interne Kommunikation in z/OS als auch für die Verbindungen mit z/OS-Partnern.
- Das Element TNSTCPIP der z/OS-Parameterdatei wird nicht mehr unterstützt, die Beschreibung ist deshalb entfallen.

### 1.4.6 Neue Funktionen, die nur im openFT Explorer zur Verfügung stehen

Die folgenden neuen Funktionen stehen nur im openFT Explorer zur Verfügung:

- Öffentliche Schlüssel exportieren

  Über das Menü *Administration*, Befehl *Schlüsselverwaltung - Öffentlichen Schlüssel exportieren…* kann der FT-Verwalter öffentliche Schlüssel der lokalen Instanz exportieren.

- Diagnoseinformationen und Konsolmeldungen löschen

  Über das Menü *Administration*, Befehle *Diagnoseinformationen* und *Konsolmeldungen* kann der FT-Verwalter Diagnoseinformationen und Konsolmeldungen löschen.

- Das Logging steht auch im Objektbaum des openFT Explorers zur Verfügung.

Weitere Details finden Sie in der Online-Hilfe des openFT Explorers.

## 1.5 Darstellungsmittel

In diesem Handbuch werden folgende Darstellungsmittel verwendet:

`dicktengleiche Schrift`
Dicktengleiche Schrift wird für Eingaben und Beispiele verwendet.


für Hinweistexte.


für Warnhinweise.

## 1.6 Readme-Datei

Funktionelle Änderungen und Nachträge der aktuellen Produktversion zu diesem Handbuch entnehmen Sie bitte gegebenenfalls den produktspezifischen Readme-Dateien.

Readme-Dateien stehen Ihnen online bei dem jeweiligen Produkt zusätzlich zu den Produkthandbüchern unter *http://manuals.ts.fujitsu.com* zur Verfügung.

## 1.7 Aktuelle Informationen im Internet

Aktuelle Informationen zur openFT-Produktfamilie finden Sie im Internet unter *http://www.fujitsu.com/de/openFT* (deutsch) bzw. *http://www.fujitsu.com/ts/openFT* (englisch).

# 2 openFT - der Managed File Transfer

Managed File Transfer ist ein Begriff, der den hohen Anspruch der openFT-Produkte dokumentiert. Hohe Ansprüche an einen unternehmensweiten File Transfer resultieren zum einen aus der heute üblichen Vielfalt hinsichtlich der installierten Hard- und Software, zum anderen aus den unterschiedlichen, den Bedürfnissen Ihres Unternehmens entsprechenden Anforderungen an die Datenübertragung selbst. Ein weiterer wichtiger Aspekt für einen unternehmensweiten File Transfer sind die von openFT gebotenen Automatisierungsmöglichkeiten und Sicherheitsfunktionen. Darüber hinaus lässt die **Zentrale Administration** eines openFT-Netzes und die Darstellung der Betriebszustände openFT zum **verwalteten File Transfer** werden.

Fujitsu Technology Solutions bietet für den Managed File Transfer die umfangreiche openFT-Produktfamilie an, mit der **heterogene Rechnersysteme** (Hardware und Software) vieler Hersteller vom Mainframe bis zum PC bedient werden können. Die openFT-Produkte können auf Betriebssystemen wie Windows, Unix-Systemen, BS2000-Systemen, z/OS und anderen eingesetzt werden.

Auch **heterogene Netze** wie z.B. TCP/IP, ISO-FTAM, X.21/X.25 und ISDN stellen für openFT kein Hindernis dar. Die ständige Integration neuer Plattformen und Netztypen garantiert auch für die Zukunft eine hohe Verfügbarkeit der openFT-Produkte. Nicht alle Netze werden auf allen Plattformen unterstützt.

Die Integration der **ISO-Norm 8571 FTAM** (File Transfer, Access and Management) garantiert einheitliche Schnittstellen für Aufträge zu openFT-Partnern und beliebigen FTAM-Partnern (nicht unter z/OS verfügbar).

Die Unterstützung des **FTP-Protokolls** ermöglicht die Anbindung zu FTP-Servern und von FTP-Clients auf beliebigen Plattformen.

Funktionen wie z.B. Auftragsspeicherung, automatischer Wiederanlauf, Job- und Dateimanagement, Folgeverarbeitung, Betriebsmittelsteuerung, Programmschnittstellen, Verschlüsselung und Authentifizierung zeigen die immense Leistungsbreite und machen die openFT-Produkte zum Managed File Transfer.

Die **Auftragsspeicherung** ermöglicht es Ihnen, die Übertragung durch **asynchronen File Transfer** gezielt zu einem gewünschten Zeitpunkt zu starten, beispielsweise um Gebühren zu sparen oder das Eintreffen bestimmter Ereignisse abzuwarten. Der **automatische Wiederanlauf** gewährleistet im Störungsfall, z.B. Netz- oder Rechnerausfall, eine konsistente Weiterführung der Datenübertragung nach Beendigung der Störung.

**Automatisierung** wird u.a. durch die Möglichkeit zur Vor-, Nach- und Folgeverarbeitung realisiert:

- Lokale oder ferne **Vorverarbeitung** ermöglicht es, innerhalb eines Sende- bzw. Empfangsauftrags Daten zu erzeugen, z.B. durch Starten eines Jobs, und diese anschließend zum fernen bzw. lokalen System zu übertragen.
- Lokale oder ferne **Nachverarbeitung** ermöglicht es, die übertragenen Daten innerhalb eines Empfangs- bzw. Sendeauftrags weiter zu verarbeiten.
- Innerhalb eines Auftrags kann sowohl eine Vor- als auch eine Nachverarbeitung durchgeführt werden.
- **Folgeverarbeitung** erlaubt den Start eines beliebigen Jobs im Anschluss an eine Dateiübertragung. Den Start der Folgeverarbeitung können Sie abhängig vom Erfolg der Übertragung definieren.

Die **Programmschnittstellen** gestatten die Implementierung von openFT-Funktionen in Programmen.

**Dateimanagement** im fernen und im lokalen System bietet z.B. die Möglichkeit, Dateieigenschaften zu verändern.

Die **Betriebsmittelsteuerung** erlaubt es, Dateiübertragungsaufträge zu einem beliebigen Zeitpunkt zu hinterlegen und erst bei Verfügbarkeit des Partnersystems automatisch ausführen zu lassen. Die Verwendung von Monitor-Jobvariablen im BS2000-System ist ebenfalls möglich.

Beim **synchronen File Transfer** müssen Sie auf das Ende der Ausführung der Datenübertragung warten und können im Anschluss daran sofort auf das Ergebnis reagieren.

Der Schutz des eigenen Datenbestandes bekommt innerhalb eines Unternehmens angesichts offener Netze einen immer höheren Stellenwert. Die in die openFT-Produkte integrierte **FTAC-Funktionalität** (optional in openFT (BS2000) und openFT (z/OS)) bietet umfangreiche und individuell skalierbare Schutzfunktionen:

– Entkopplung von Zugangs- und Login-Berechtigung
– Zugriffsrechte abhängig von Partnersystemen
– benutzerspezifische Zugriffsrechte
– flexible Abstufung der Zugriffsrechte
– Protokollierung jeder Berechtigungsprüfung

Die **Protokollierung** von Dateiübertragungsaufträgen und Berechtigungsüberprüfungen erlaubt eine Auswertung vergangener Aufträge und Zugriffe und stellt somit ein weiteres Sicherheitsmerkmal dar.

Die **Verschlüsselung** von Auftragsbeschreibungs- und Übertragungsdaten ist eine weitere Schutzstufe von openFT. Zu den Auftragsbeschreibungsdaten gehören unter anderem die Berechtigungsdaten zum Zugang und Zugriff auf Daten (z.B. Zugangsberechtigung, Dateikennwort). Außerdem besteht die Anschlussmöglichkeit an System-Security-Funktionen wie z.B. SECOS auf BS2000, RACF und ACF2 auf z/OS.

Für Aufträge mit openFT-Partnern wird eine erweiterte Identitätsprüfung des Kommunikationspartners, die sogenannte **Authentifizierung**, angeboten. Sie basiert auf der Adressierung über netzweit eindeutige Identifikationen für openFT-Instanzen und dem Austausch partnerspezifischer Schlüsselinformationen.

Die openFT-Script-Schnittstelle ermöglicht bei einer großen Anzahl von Dateien (z.B. bei ganzen Verzeichnisbäumen) eine **wiederanlaufgesicherte Übertragung**, d.h. wenn das Netz oder der Rechner bei der 258ten Datei ausfällt, setzt openFT-Script nach dem Wiederanlauf die Übertragung genau an dieser Stelle fort. Die openFT-Script-Schnittstelle steht auf Unix- und Windows-Systemen zur Verfügung.

## 2.1 Heterogene Rechnersysteme

Eine Stärke der openFT-Produkte ist die Möglichkeit, unterschiedliche Rechner, insbesondere Rechner mit unterschiedlichen Betriebssystemen und von verschiedenen Herstellern, zu koppeln. Voraussetzung für die Datenübertragung zwischen zwei Rechnern ist, dass eine Transportverbindung zwischen diesen beiden Rechnern existiert und dass auf den beteiligten Rechnern eines der openFT-Produkte, ein FTAM-Produkt (nicht auf z/OS) oder eine FTP-Anwendung installiert ist. FTAM steht auf z/OS nicht zur Verfügung.

Die openFT-Produkte sind aufeinander abgestimmt. Sie übertragen Dateien unter Beachtung der Dateistrukturen und der Dateieigenschaften. Andererseits können sich die openFT-Produkte nicht über die Konventionen hinwegsetzen, die für das Betriebssystem gelten. Für die richtige Zeichendarstellung bei der Datenübertragung zwischen bestimmten Betriebssystemen ist ggf. eine Datenkonvertierung notwendig.

### 2.1.1 Datenkonvertierung

Die Codierung, d.h. die systeminterne Darstellung einzelner Zeichen, Buchstaben und Ziffern, ist betriebssystemabhängig. Daher müssen die Daten ggf. konvertiert werden, denn

– Unix- und Windows-Rechner verwenden intern einen ASCII-basierten Code (American Standard Code for Information Interchange). Für Unix-Systeme ist dies ein ISO-8859-x-Code, der in der ISO-Norm 8859 beschrieben ist, für Windows-Systeme ist dies ein von Microsoft definierter Code wie z.B. der Zeichensatz CP1252 für Westeuropa mit Euro-Zeichen.

– BS2000-Systeme und z/OS-Rechner hingegen verwenden intern in der Regel einen EBCDIC-Code (Extended Binary-Coded Decimal Interchange Code).

Datenkonvertierung zwischen openFT-Partnern betrifft immer die Zeichen, aus denen übertragene Parameterwerte (wie Dateinamen, Benutzerkennungen, Folgeverarbeitungs-Strings usw.) bestehen.

Die Konvertierung des Dateiinhalts ist dagegen nur bei Übertragung von Dateien im Textformat relevant, bei Übertragung in anderen Dateiformaten (binär, transparent, ...) konvertiert openFT die Daten nicht.

Achten Sie darauf, dass die openFT-Partner Codes mit dem gleichen Zeichenvorrat verwenden. Ist dies nicht der Fall, werden eventuell nicht alle Zeichen einer Textdatei (z.B. Umlaute) korrekt im Zielsystem dargestellt. Wenn Sie Dateien mit openFT-Partnern ab V10 übertragen, können Sie die lokal und remote für die Datenkonvertierung zu verwendenden "Coded Character Sets" im Auftrag zuweisen. Mit diesen Partnersystemen können auch Unicode-Dateien übertragen werden, siehe Abschnitt „Übertragung von 7 Bit-, 8 Bit- und Unicode-Dateien“ auf Seite 81.

### 2.1.2 openFT-Produktfamilie

Eine Übersicht der openFT-Produktfamilie, die Ihnen Auskunft über die für Ihre Rechner angebotenen openFT-Produkte gibt, finden Sie in den folgenden Tabellen.

**openFT-Produktfamilie**

| Produkt | Betriebssystem | Bemerkung |
|---|---|---|
| openFT (Unix-Systeme) | AIX, Linux, HP-UX, Oracle Solaris | Intel x86-Architektur und IBM-Prozessoren, weitere Systeme auf Anfrage |
| openFT (Windows) | Microsoft Windows | x86-Architektur |
| openFT (BS2000) | BS2000 | BS2000-Systeme von Fujitsu Technology Solutions |
| openFT (z/OS) | z/OS | z/OS-Systeme von IBM |

### openFT-Zusatzprodukte

| Produkt | Betriebssystem | Bemerkung |
|---|---|---|
| openFT-FTAM für Unix-Systeme | AIX, Linux, HP-UX, Oracle Solaris | Intel x86-Architektur und IBM-Prozessoren, weitere Systeme auf Anfrage |
| openFT-FTAM für Windows-Systeme | Microsoft Windows | x86-Architektur |
| openFT-FTAM für BS2000-Systeme | BS2000 | FTAM-Funktionalität für BS2000-Systeme von Fujitsu Technology Solutions |
| openFT-FTP für Unix-Systeme | AIX, Linux, HP-UX, Oracle Solaris | Intel x86-Architektur und IBM-Prozessoren, weitere Systeme auf Anfrage |
| openFT-FTP für Windows-Systeme | Microsoft Windows | x86-Architektur |
| openFT-FTP für BS2000-Systeme | BS2000 | FTP-Funktionalität für BS2000-Systeme |
| openFT-FTP für z/OS | z/OS | FTP-Funktionalität für z/OS-Systeme |
| openFT-AC für BS2000-Systeme | BS2000 | FTAC-Funktionalität für BS2000-Systeme |
| openFT-AC für z/OS | z/OS | FTAC-Funktionalität für z/OS-Systeme |
| openFT-CR | alle Plattformen der openFT-Produktfamilie | Datenverschlüsselung (exportbeschränkt) |

## 2.2 Heterogene Netze

Ein Verbund von Rechnern und anderen Geräten wird als Netz bezeichnet. Wenn Rechner mit gleichartigen Kommunikationsarchitekturen miteinander gekoppelt werden, spricht man von einem homogenen Netz.

Als heterogenes Netz wird ein Rechnerverbund bezeichnet, in dem Rechner mit verschiedenen Kommunikationsarchitekturen miteinander kommunizieren. Wichtige Eigenschaften von Rechnernetzen sind die zu überbrückende Entfernung, die Art der Übertragungsstrecke, die Nutzung von öffentlichen Diensten, die Übertragungsgeschwindigkeit und die Art der Protokolle, d.h. die Gesamtheit der Regeln und Vorschriften, die bei der Informationsübertragung beachtet werden müssen.

Die bekanntesten von openFT unterstützten Netze sind TCP/IP, ISO, SNA, X.21/X.25, ISDN. Nicht alle Netztypen werden auf allen Plattformen unterstützt.

Netzmanagement in heterogenen Netzen basiert in den meisten Fällen auf **SNMP** (Simple Network Management Protocol).

Die openFT-Produkte auf BS2000 und Windows unterstützen das SNMP-basierte Netzmanagement und unterstreichen so ihre Stellung in offenen Netzen.

### 2.2.1 OSI-Referenzmodell

Um Daten austauschen zu können, müssen sich Rechner miteinander verständigen können. Kommunikation ist nur dann möglich, wenn die an der Kommunikation beteiligten Rechner für den Datenaustausch die gleichen Datenformate verwenden und sich an vereinbarte Vorgehensweisen bei der Datenübertragung halten. Die Summe der Verhaltensregeln und Datenformate für die Kommunikation wird als Protokoll bezeichnet. Die Definition von Protokollen erfolgt einerseits durch Hersteller (beispielsweise die openFT-Protokolle), andererseits haben sich Gremien etabliert, die herstellerunabhängige Protokolle definieren. ISO (International Organization for Standardization) stellt mit dem OSI-Referenzmodell (**O**pen **S**ystems **I**nterconnection) das bekannteste Modell zur Kommunikationsarchitektur und die umfassendste Protokollsammlung zur Verfügung.

Das OSI-Modell strukturiert die Kommunikationsfunktionen von Rechnersystemen und schafft die Basis für die Normung der Protokolle und Dienste. Es schreibt fest, welche Funktionen die an der Kommunikation beteiligten Komponenten erbringen müssen.

Das OSI-Referenzmodell besteht aus sieben hierarchisch aufeinanderliegenden Schichten. Jeder Schicht sind im Rahmen der Kommunikationsaufgaben spezifische Funktionen zugeordnet.

| Schichten | Bezeichnung | Funktionen | |
|---|---|---|---|
| Schicht 7 | Anwendungsschicht (Application Layer) | Koordiniert und steuert die Durchführung von Kommunikationsaufgaben für eine Anwendung | **A** N W E N D U N G |
| Schicht 6 | Darstellungsschicht (Presentation Layer) | Regelt die Form der Informationsdarstellung und ermöglicht damit eine benutzer- und geräteunabhängige Kommunikation | |
| Schicht 5 | Kommunikationssteuerungsschicht (Session Layer) | Regelt den Ablauf der Kommunikation | |
| Schicht 4 | Transportschicht (Transport Layer) | Regelt den zuverlässigen Datenaustausch zwischen zwei kommunizierenden Partnern | **T** R A N S P O R T |
| Schicht 3 | Vermittlungsschicht (Network Layer) | Regelt den Datenaustausch zwischen zwei Endsystemen (Rechnern) | |
| Schicht 2 | Sicherungsschicht (Data Link Layer) | Sichert die Übertragung auf den einzelnen Teilstrecken des gesamten Übertragungsweges (Prozeduren) | |
| Schicht 1 | Bitübertragungsschicht (Physical Layer) | Stellt die rein physikalische Verbindung her (über das für die Übertragung benutzte Medium) | |

OSI-Referenzmodell

Die einzelnen Schichten nehmen den Dienst der jeweils darunter liegenden Schicht in Anspruch und bieten der darüberliegenden Schicht ebenfalls einen genau definierten Dienst. Nur die Bitübertragungsschicht muss ihren Dienst zusammen mit dem physikalischen Medium selbst erbringen. Die aktiven Elemente innerhalb einer Schicht, die die Funktionen erbringen, werden Instanzen genannt.

Jede Schicht wird festgelegt durch den Dienst, den sie erbringt, und die Dienste, die sie von der darunter liegenden Schicht in Anspruch nimmt. Bei der Kommunikation arbeiten die verschiedenen Rechner auf der jeweils gleichen Schicht über gemeinsame Protokolle zusammen.

Die Funktionalität jeder Schicht im OSI-Referenzmodell kann in der Regel von verschiedenen Protokollen erbracht werden. Für die Kommunikation ist entscheidend, dass die direkten Partnerinstanzen für eine bestimmte Aufgabe dieselben Protokolle verwenden. Zu diesem Zweck werden Profile definiert.

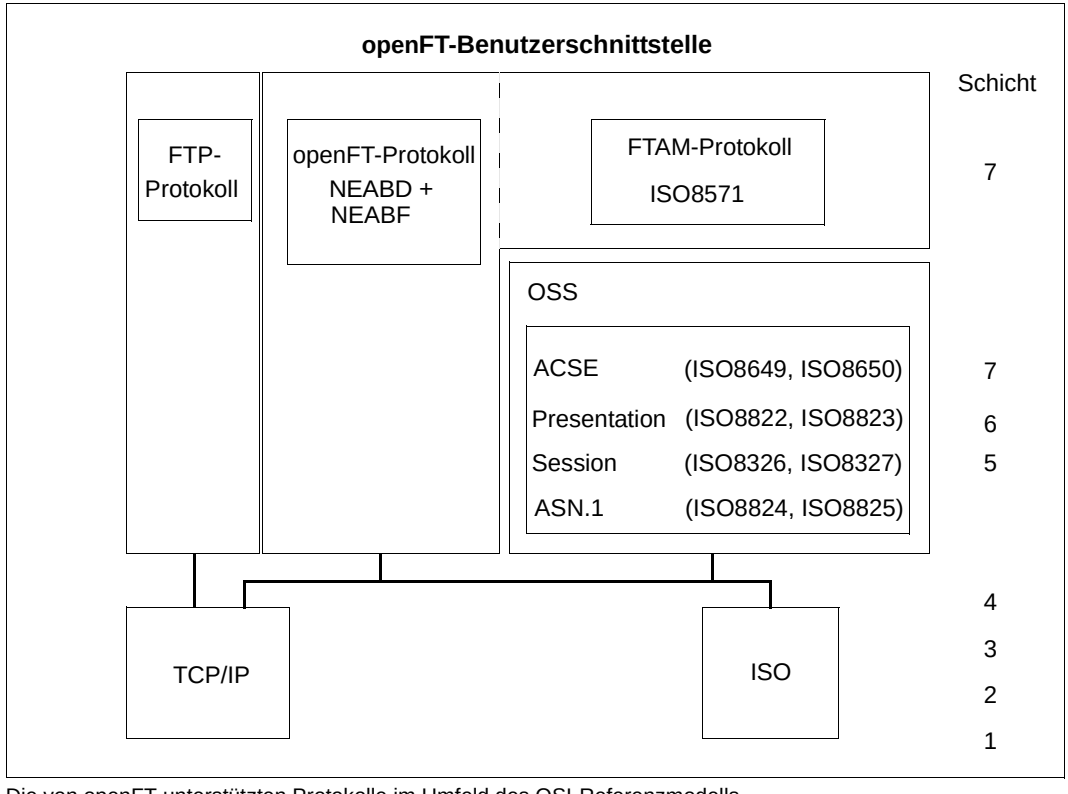
Unter einem Profil versteht man eine genaue Festlegung, welche Protokolle oder welche Protokollvariante auf welcher Schicht zur Erfüllung einer bestimmten Aufgabe eingesetzt werden sollen. Profile werden von nationalen oder internationalen Organisationen oder Interessenverbänden festgelegt.

### 2.2.2 Einordnung der openFT-Produktlinie in das OSI-Referenzmodell

Die openFT-Produkte gehören in die Anwendungsschichten (Schicht 5 - 7) des OSI-Referenzmodells. Sie unterstützen das genormte openFT-Protokoll, das von ISO genormte FTAM-Protokoll ISO8571 sowie das durch RFC959 definierte File Transfer Protokoll (FTP).

Die openFT-Produkte können unterschiedliche Transportsysteme mit unterschiedlichen Transportprotokollen nutzen.

Die folgende Grafik zeigt, welche Kombinationen von Anwendungs- und Transportprotokollen bei der Dateiübertragung möglich sind:




Die von openFT unterstützten Protokolle im Umfeld des OSI-Referenzmodells

openFT (z/OS) unterstützt für die Dateiübertragung nur das openFT-Protokoll und das FTP-Protokoll.

Eine Übersicht der Transportsysteme und -protokolle, die den Betrieb von openFT-Produkten erlauben, entnehmen Sie bitte den jeweiligen Produktblättern.

### 2.2.3 openFT-Partner

openFT kann Dateiübertragung und Dateimanagement mit Partnersystemen betreiben, die in den Anwendungsschichten die openFT-Protokolle NEABD und NEABF unterstützen.

Diese Partnersysteme werden im Folgenden openFT-Partner genannt. openFT-Partner können auf Mainframe-Plattformen (BS2000-Systeme, z/OS) sowie auf offenen Plattformen (Unix-Systeme, Windows-Systeme) ablaufen.

Dabei können abhängig von der jeweiligen Transportsystem-Software verschiedene Transportprotokolle benutzt werden:

– TCP/IP-Transportprotokolle (alle Plattformen)
– SNA-Protokolle (z/OS)
– ISO-Transportprotokolle (BS2000-Systeme, Unix- und Windows-Systeme)
– X.25-Transportprotokolle (Windows- und Linux-Systeme)

Der Funktionsumfang ist für eine bestimmte openFT-Version auf den unterschiedlichen Plattformen weitgehend identisch, kleinere Unterschiede sind durch das jeweilige Betriebssystem bedingt.

Diese Protokolle, auch FTNEA-Protokolle genannt, wurden offengelegt, so dass es auch Produkte von anderen Herstellern gibt, die diese Protokolle unterstützen.

### 2.2.4 FTAM-Partner

Mit der in openFT gebotenen FTAM-Erweiterung kann openFT Dateiübertragung und Dateimanagement mit Partnersystemen betreiben, die in den Schichten 5 - 7 des OSI-Referenzmodells ISO-Protokolle unterstützen. Im Folgenden werden diese Partnersysteme FTAM-Partner genannt. In der internationalen Norm ISO 8571 (FTAM, File Transfer, Access and Management) sind die Protokolle für diese Art der Dateiübertragung festgelegt.

BS2000-Systeme benötigen zusätzlich das Softwarepaket OSS zur Realisierung der Schichten 5 - 7.

FTAM steht auf z/OS nicht zur Verfügung.

#### Implementierung der FTAM-Normen in openFT

Aus dem vollständigen Funktionsumfang der Basisnormen wurde eine Untermenge ausgewählt gemäß der internationalen und europäischen Profile ISO/EN ISP 10607-3 und ISO/EN 10607-6. Diese Funktions-Vornormen sind wiederum mit anderen Funktionsnormen (auch Implementation Agreements) abgestimmt, zum Beispiel mit den entsprechenden Implementation Agreements des IGOSS in Nordamerika und entsprechenden Profilen in Asien und Australien.

ENV 41204 und ENV 41205 sind die alten, aber noch gebräuchlichen Bezeichnungen für EN 10607-3 und EN 10607-6 und sind inhaltlich identisch mit den von ISO verabschiedeten internationalen Profilen ISO/IEC ISP 10607-3 (1990) sowie ISO/IEC ISP 10607-6 (1990). EN 10607-3 und EN 10607-6 enthalten zusätzlich europäische Character Sets.

Mit diesen Profilen wird beispielsweise festgelegt, mit welchen Dateiattributen tatsächlich gearbeitet wird und welche Operationen mit diesen Attributen erlaubt sind, unabhängig davon, welches Betriebssystem verwendet wird. Zur betriebssystemübergreifenden Darstellung wird ein **virtueller Dateispeicher** (Virtual Filestore) genutzt, in den der Inhalt des realen Speichers mit einer normgerechten Darstellung der Dateiattribute übertragen wird. Das Umsetzen der Dateiattribute im Betriebssystem auf FTAM-Norm bzw. umgekehrt ist Bestandteil der FTAM-Funktionalität. Es gibt dabei drei Gruppen von Dateiattributen: Kernel group, Storage group und Security group (siehe Seite 107).

Die Erfüllung der FTAM-Norm bedingt auch gewisse Einschränkungen gegenüber der von openFT-Protokollen gebotenen Funktionsvielfalt. Die Übertragung von Folgeverarbeitungsdaten zu FTAM-Partnern ist protokollbedingt nicht möglich.

Die Abbildungsmechanismen zwischen dem realen Dateispeicher und dem virtuellen Dateispeicher sind auf Seite 107 detailliert beschrieben.

### 2.2.5 FTP-Partner

Bei Nutzung des FTP-Protokolls ist die Kommunikation nur über TCP/IP möglich. Darüber hinaus gibt es bei der Bedienung von FTP-Servern gegenüber openFT-Partnern eine Reihe von Besonderheiten, die überwiegend durch Einschränkungen des FTP-Protokolls bedingt sind, z.B.:

- Es gibt keinen Wiederanlauf.
- Verschlüsselung ist nur für Outbound-Aufträge zu einem FTP-Server mit Secure-FTP-Unterstützung mit dem Protokoll TLS möglich. Dazu muss openFT-Crypt (Liefereinheit openFT-CR) installiert sein.
- Folgeverarbeitung und Coded Character Sets werden nur lokal unterstützt.
- Einige Eigenschaften einer Datei werden nicht übertragen. Details siehe jeweiliges Handbuch „Kommandoschnittstelle“.

Bitte beachten Sie, dass die weiteren openFT-Funktionen (Vor- und Nachverarbeitung, FTAC, ...) nur genutzt werden können, wenn auf dem System, wo die Vor- und Nachverarbeitung ausgeführt werden soll, openFT als FTP-Server eingesetzt wird.

## 2.3 Dateien übertragen

Die wesentliche Aufgabe von openFT ist die Übertragung von Dateien zwischen zwei Partnersystemen. Dazu stellen Sie in Ihrem System - dem lokalen System - einen Auftrag zur Dateiübertragung. Mit diesem Auftrag können Sie eine Datei an ein Partnersystem (fernes System) senden oder eine Datei von einem Partnersystem holen. Ein Partnersystem kann auch an Ihr System Dateien senden oder Dateien von Ihrem System holen.

Aufträge, die Sie in Ihrem lokalen System stellen, werden als **Outbound-Aufträge** (nach außen gerichtet) bezeichnet. Aufträge, bei denen die Initiative im fernen System liegt, werden als **Inbound-Aufträge** (nach innen gerichtet) bezeichnet.

In einem Dateiübertragungsauftrag können Sie angeben, ob die zu übertragende Datei eine reine Textdatei ist oder unstrukturierte bzw. strukturierte Binärdaten enthält. Danach richtet sich die Behandlung der Daten bei der Übertragung, siehe Abschnitt „Datenkonvertierung“ auf Seite 24. Eine Sonderrolle spielt das sogenannte "transparente" Dateiformat, mit dem Sie BS2000-Dateien mit allen Eigenschaften ohne Umcodierung im Empfangssystem speichern können. Dies ist z.B. dann erforderlich, wenn ein Windows- oder Unix-System zur Verteilung von BS2000-Software verwendet wird.

Auf Unix- und Windows-Systemen ist es auch möglich, mit einem Auftrag komplette Verzeichnisse an ein Partnersysstem zu senden.

Für alle Dateiübertragungsaufträge mit openFT-Partnern kann eine Vorverarbeitung, Nachverarbeitung und/oder Folgeverarbeitung vereinbart werden. Eine Folgeverarbeitung können Sie sowohl im lokalen als auch im fernen System für eine erfolgreiche und auch für eine misslungene Übertragung vorgeben. Was Sie mit Vor-, Nach- bzw. Folgeverarbeitung alles machen können, finden Sie im Abschnitt „Dateiübertragung mit Vorverarbeitung, Nachverarbeitung und Folgeverarbeitung“ auf Seite 38.

Solange die Übertragung einer Datei nicht vollständig beendet ist, sollten Sie diese Datei nicht anderweitig bearbeiten, da sie sonst inkonsistent werden kann. Im BS2000 sind DMS-Dateien während der gesamten Auftragsdurchführung geschützt.

Sie können entscheiden, wann openFT Ihren Übertragungsauftrag ausführen soll: entweder sofort oder zu einem Zeitpunkt, den Sie selbst festlegen können. Wenn Sie einen synchronen Auftrag stellen, führt ihn openFT immer sofort aus. Soll ein Auftrag später ausgeführt werden, müssen Sie einen asynchronen Auftrag starten und dabei die Ausführungszeit angeben.

**Komprimierte Übertragung**

Sie können für jeden Auftrag angeben, ob die Datei komprimiert übertragen und welche Komprimierungsart verwendet werden soll (Byte-Komprimierung oder Zip-Komprimierung).

Damit können Sie

- Übertragungszeiten verkürzen
- Übertragungswege entlasten
- Übertragungskosten sparen

### 2.3.1 Startzeitpunkt der Übertragung festlegen

Wenn Sie einen **synchronen Auftrag** gestartet haben, wird die Datei sofort übertragen. Sie haben den Vorteil, dass Sie sofort wissen, ob die Übertragung erfolgreich war. Auf Unix- und Windows-Systemen informiert Sie während der gesamten Übertragungszeit eine Anzeige über den Fortschritt der Übertragung.

Das Ergebnis können Sie unmittelbar als Entscheidungskriterium für weitere Schritte nutzen. Ist die Übertragung nicht möglich, weil z.B. der Partner nicht erreichbar ist, wird die Übertragung abgebrochen und Sie können den Auftrag zu einem späteren Zeitpunkt neu starten.

Beim **asynchronen Auftrag** überträgt openFT die Datei entweder zum nächstmöglichen oder zu einem von Ihnen vorgegebenen Zeitpunkt. Damit kann die Übertragung zu einem Zeitpunkt starten, zu dem z.B. der Partner erreichbar ist oder die Gebühren für die Übertragung besonders günstig sind. Der Auftrag wird in einem **Auftragsbuch** gespeichert und Sie erhalten eine Bestätigung über die Auftragsannahme. Ihr System ist damit sofort wieder frei für andere Aufgaben und Sie müssen sich nicht mehr um die Auftragsausführung kümmern. Denn wenn z.B. für die Übertragung keine Verbindung aufgebaut werden kann, versucht openFT in bestimmten Zeitabständen, die Übertragung erneut zu starten, und selbst wenn während der Übertragung eine Störung auftritt, erfolgt automatisch ein Wiederanlauf. Der Wiederanlauf kann jedoch auch für asynchrone Dateiübertragungs-Aufträge partnerspezifisch oder global deaktiviert werden.

Sie können auch mehrere asynchrone Aufträge starten. Die Aufträge werden solange im Auftragsbuch geführt, bis sie ausgeführt sind, von Ihnen abgebrochen werden oder ihre global eingestellte maximale Lebensdauer erreicht ist (siehe Abschnitt „Lebensdauer eines Auftrags steuern“ auf Seite 34). Im Auftragsbuch können Sie sich jederzeit über alle noch nicht ausgeführten Aufträge informieren.

Aufträge, die ein fernes System stellt, also Inbound-Aufträge, werden von openFT im lokalen System immer als asynchrone Aufträge ausgeführt.

### 2.3.2 Lebensdauer eines Auftrags steuern

Ein asynchroner openFT-Auftrag bleibt so lange im Auftragsbuch, bis der Auftrag vollständig ausgeführt ist, er explizit gelöscht wird oder seine global eingestellte maximale Lebensdauer erreicht ist.

Sie können jedoch beim Stellen eines asynchronen Auftrags einen Zeitpunkt angeben, zu dem der Auftrag gelöscht bzw. die Übertragung abgebrochen werden soll (Cancel-Timer). Damit können Sie z.B. verhindern, dass wegen zeitweilig nicht erreichbarer Partner oder aufgrund von Netzproblemen Betriebsmittel unnötig lange belegt werden.

### 2.3.3 Auftragsbuch

Im Auftragsbuch werden alle asynchronen Dateiübertragungsaufträge gespeichert, die noch nicht ausgeführt sind. Sie können sich das Auftragsbuch jederzeit am Bildschirm anzeigen lassen. Es enthält für jeden Auftrag unter anderem folgende Informationen:

- die Übertragungsrichtung
- den Betriebszustand des Auftrags
- die Anzahl bereits übertragener Bytes
- den Initiator des Auftrags
- den lokalen Dateinamen, bei Outbound-Aufträgen auch den fernen Dateinamen
- das beteiligte Partnersystem
- Folgeverarbeitung
- Diagnoseinformationen

Der Byte-Zähler im Auftragsbuch wird in regelmäßigen Abständen aktualisiert, so dass Sie sich über den Fortschritt einer Dateiübertragung informieren können. Sie können Aufträge löschen und die Reihenfolge der Aufträge im Auftragsbuch ändern (Prioritätensteuerung).

Informationen über bereits abgeschlossene Aufträge erhalten Sie über die Logging-Funktion (siehe Seite 48).

#### Prioritätensteuerung

Die Aufträge werden nach dem FIFO-Prinzip abgearbeitet, d.h. der zuerst angestoßene Auftrag wird auch als erster bearbeitet (FIFO = First In First Out). Es gibt mehrere Prioritätsklassen, wodurch Sie die Bearbeitung eines Auftrags wie folgt steuern können:

- indem Sie beim Stellen des Auftrags die Priorität explizit angeben
- indem Sie die Priorität eines Auftrags im Auftragsbuch ändern
- indem Sie die Warteschlange des Auftragsbuchs ändern, d.h. Aufträge an die erste oder letzte Stelle in der Reihe gleichpriorer Aufträge setzen

Zusätzlich kann der FT-Verwalter Partner priorisieren, siehe Seite 57.

### 2.3.4 Automatischer Wiederanlauf

Für den Fall, dass die Übertragung einer Datei aus irgend einem Grund unterbrochen wird, bietet Ihnen openFT einen zuverlässigen Wiederanlauf. Damit sind z.B. Netzprobleme für openFT kein Hindernis, denn openFT sorgt automatisch für die Fortsetzung der Übertragung, sobald dies wieder möglich ist.

Grundlage für den automatischen Wiederanlauf sind die Auftragsspeicherung im Auftragsbuch und die sogenannten Wiederanlaufpunkte. Das sind Sicherungspunkte, mit denen sich die beiden Partnersysteme während einer Übertragung in regelmäßigen Zeitabständen synchronisieren. Wird die Übertragung unterbrochen, dann wird ab dem letzten Sicherungspunkt mit der Übertragung fortgefahren, sobald dies wieder möglich ist. Dadurch können Sie sicher sein, dass bei der Übertragung der Datei nichts verloren geht und nichts hinzugefügt wird.

Durch den festen zeitlichen Abstand der Sicherungspunkte werden bei schnellen Leitungen nicht unnötig viele Sicherungspunkte gesetzt und bei langsamen Leitungen wird der Abstand nicht zu groß.

Bei der Übertragung von Verzeichnissen wird kein automatischer Wiederanlauf durchgeführt, wenn ein zu übertragendes Verzeichnis ein Unterverzeichznis enthält.

#### Automatischen Wiederanlauf deaktivieren

In manchen Situationen ist es sinnvoll, für asynchrone Dateiübertragungs-Aufträge die Wiederanlauf-Funktion für das openFT- und FTAM Protokoll zu deaktivieren. Wenn zum Beispiel eine openFT-Initiator-Instanz zu einem Lastverteiler koppelt, der lastabhängig einen von vielen Respondern auswählt, besteht keinerlei Garantie, dass beim Wiederanlauf dieser Lastverteiler dieselbe Responder-Instanz auswählt, die anfangs den Auftrag bearbeitet hatte. Der Wiederanlauf würde scheitern, und es bliebe im ursprünglichen Responder eine „Auftragsleiche“ zurück.

Die partnerspezifische Deaktivierung des Wiederanlaufs wirkt auf Outbound-Aufträge.

Der Wiederanlauf kann auch global per Betriebsparameter, getrennt für Outbound- und Inbound-Aufträge, deaktiviert werden.

## 2.4 Dateimanagement

Zusätzlich zur Dateiübertragung bietet openFT die Möglichkeit, Dateien im lokalen und im fernen System zu "managen". Sie können Dateimanagement-Aktionen sowohl mit openFT-Anweisungen als auch als Verarbeitung innerhalb eines Dateiübertragungsauftrages ausführen. Sinnvoll ist es z.B., wenn Sie mit Dateimanagementaufträgen vor einer Dateiübertragung im fernen System die für die Übertragung oder deren Folgeverarbeitung notwendigen Bedingungen herstellen.

Außerdem können dadurch z.B. von Windows- oder Unix-Systemen aus lokale und ferne Systeme über eine komfortable, dem Windows Standard nachempfundene Oberfläche gesteuert werden, ohne dass der Benutzer die Syntax des fernen Systems kennen muss.

Folgende Aktionen können Sie mit dem Dateimanagement durchführen:

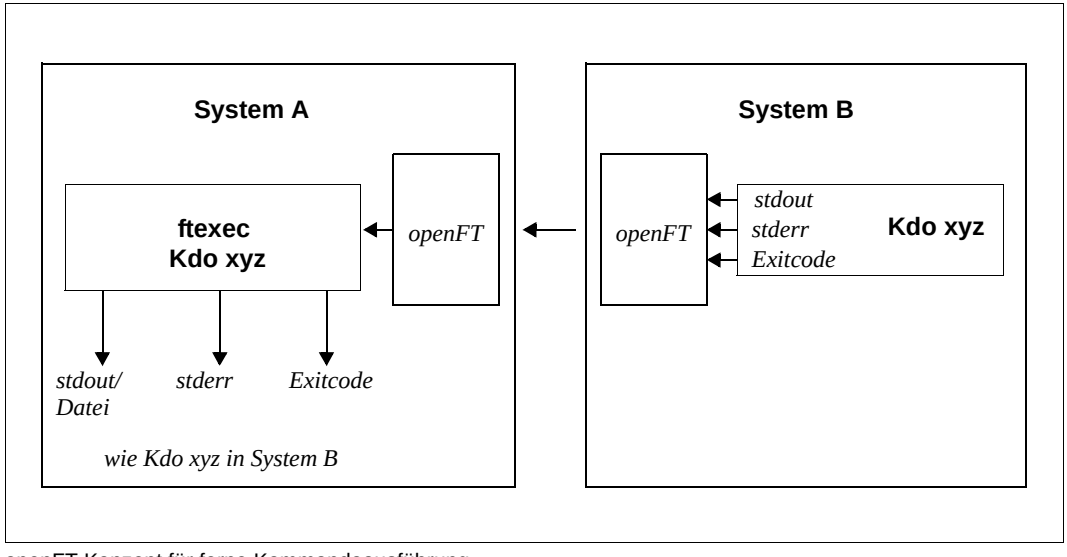
- Dateien umbenennen
- Dateien löschen
- Dateiattribute abfragen, z.B. die Größe einer Datei
- Dateiattribute ändern, z.B. die Zugriffsrechte
- Verzeichnisse anzeigen
- Verzeichnisse erzeugen
- Verzeichnisse umbenennen
- Verzeichnisse löschen

## 2.5 Ferne Kommandoausführung

openFT ermöglicht die Ausführung von Betriebssystemkommandos auf fernen Rechnern (ftexec-Mechanismus). Dabei werden der Beendigungscode (Exitcode) und die Ausgaben des Kommandos auf stdout und stderr (entspricht SYSLST und SYSOUT in BS2000) so zurückgeliefert, als würde das Kommando auf dem lokalen Rechner ausgeführt.

Damit wird es möglich, ferne Kommandos transparent in lokale Kommandoprozeduren einzubinden.

Die nachfolgende Grafik verdeutlicht das Konzept der fernen Kommandoausführung.




openFT-Konzept für ferne Kommandoausführung

## 2.6 Automatisierung

openFT ermöglicht Ihnen Jobmanagement-Funktionen wie Dateiübertragung mit Vor-/ Nach- und Folgeverarbeitung, Verwendung von Monitor-Jobvariablen im BS2000 sowie die Nutzung von File-Transfer-Funktionen in Dialogprozeduren und über Programmschnittstellen.
Unterstützt wird die Automatisierung auch durch die Möglichkeit, den Startzeitpunkt und die Lebensdauer von Aufträgen zu steuern, siehe entsprechende Abschnitte.
Die Erzeugung eindeutiger Dateinamen durch die Nutzung von openFT-Variablen erleichtert das Anwendungsdesign und reduziert Abstimmungsaufwände.

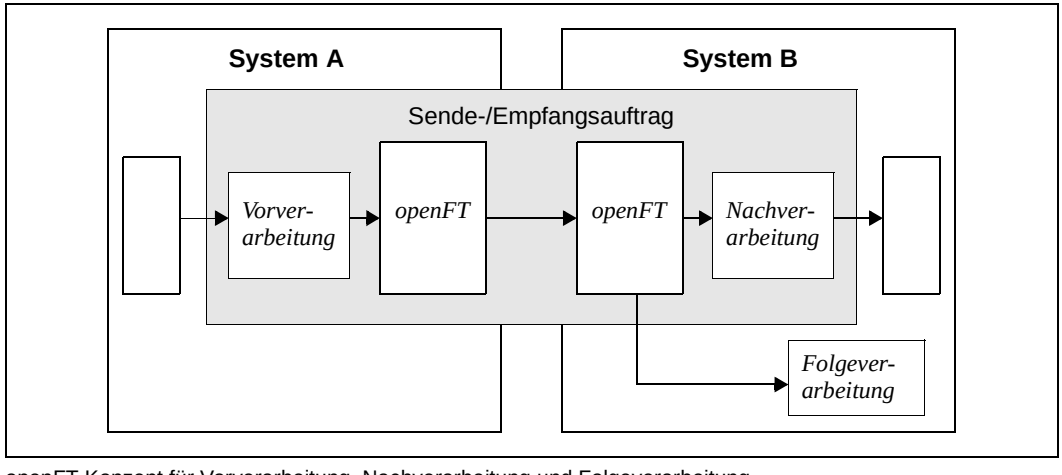
### 2.6.1 Dateiübertragung mit Vorverarbeitung, Nachverarbeitung und Folgeverarbeitung

Für eine Dateiübertragung können Sie festlegen,

– ob innerhalb des Auftrags eine Vorverarbeitung oder eine Nachverarbeitung durchgeführt werden soll. Vorverarbeitung im Sendesystem und Nachverarbeitung im Empfangssystem sind immer möglich und können innerhalb eines Auftrags auch kombiniert werden.

– ob im Anschluss an die Übertragung eine Folgeverarbeitung durchgeführt werden soll. Folgeverarbeitung können Sie sowohl für erfolgreiche wie auch für misslungene Übertragung für das lokale und ferne System festlegen.

Die nachfolgende Grafik verdeutlicht das Konzept einer Dateiübertragung mit Vor-/ Nach- und Folgeverarbeitung.




openFT-Konzept für Vorverarbeitung, Nachverarbeitung und Folgeverarbeitung

Vor- und Nachverarbeitung finden immer innerhalb des openFT-Auftrags, Folgeverarbeitung im Anschluss an den Auftrag statt.

Um zu vermeiden, dass in einer Verarbeitung Betriebsmittel unnötig blockiert werden (z.B. durch Deadlocks oder hängende Prozesse), sollten Aufträge gegebenenfalls mit einem Abbruchzeitpunkt versehen werden.

#### 2.6.1.1 Vorverarbeitung und Nachverarbeitung

Bei der Vorverarbeitung können Sie innerhalb eines Dateiübertragungsauftrags **vor** der Übertragung die Sendedaten bearbeiten. Das können Betriebssystemkommandos, Programmaufrufe oder Prozeduraufrufe sein, um die Daten vor der Übertragung zu erzeugen oder zu bearbeiten. Die Kommandos können zum Beispiel Informationen aus einer großen Datenbasis extrahieren (Datenbankabfrage) oder Daten aufbereiten (Komprimieren, Verschlüsseln), um sie dann an openFT zur Dateiübertragung zu übergeben.

Bei der Nachverarbeitung können Sie innerhalb eines Dateiübertragungsauftrags **nach** der Übertragung die Empfangsdaten durch ein oder mehrere Kommandos bearbeiten. Das können Betriebssystemkommandos, Programmaufrufe oder Prozeduraufrufe sein. Die Kommandos können zum Beispiel mit externen Routinen verschlüsselte oder komprimierte Daten entschlüsseln/dekomprimieren.

openFT-Aufträge mit ferner Vor- oder Nachverarbeitung können auch von älteren FT- oder openFT-Versionen abgegeben werden. Entscheidend ist, dass im fernen System eine openFT-Version eingesetzt wird, die die Vor- bzw. Nachverarbeitung unterstützt.

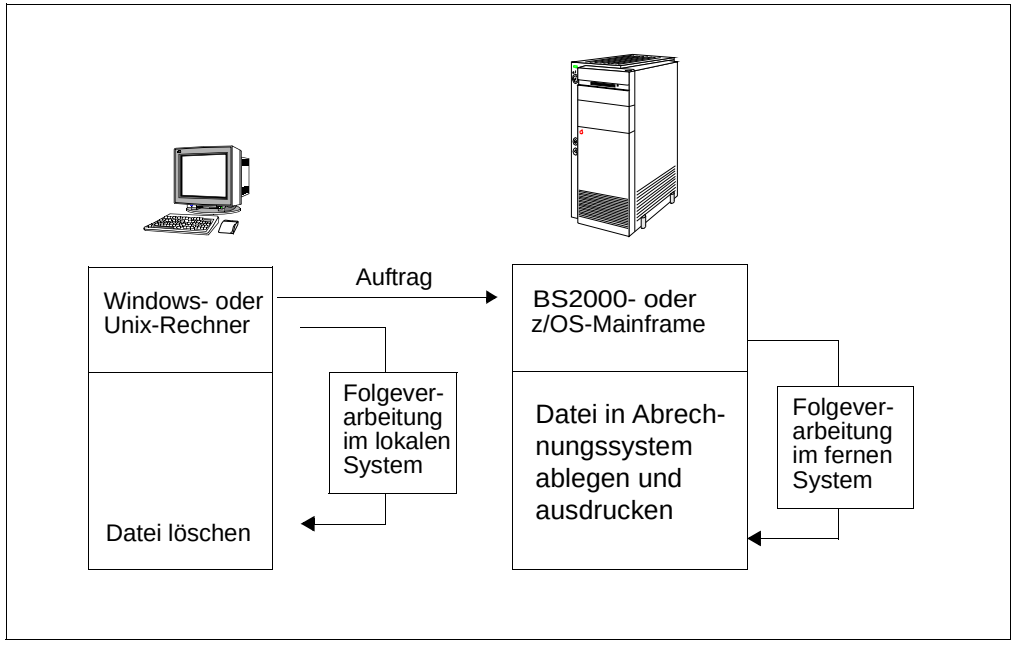
#### 2.6.1.2 Folgeverarbeitung

Die von openFT angebotene Option "Folgeverarbeitung" versetzt Sie in die Lage, abhängig vom positiven oder negativen Dateiübertragungsergebnis, im lokalen und/oder fernen System Anweisungen oder Kommandos abarbeiten zu lassen. Geben Sie eine Folgeverarbeitung für das ferne System an, müssen Sie dabei die Syntax des dort vorhandenen Betriebssystems verwenden. Für die Verwendung in Kommandos stellt openFT Variablen zur Verfügung, die bei der Ausführung des Kommandos durch die Werte aus dem Übertragungsauftrag ersetzt werden.

Bei der Übertragung von Verzeichnissen wird die Folgeverarbeitung für jeden einzelnen Dateiübertragungsauftrag ausgeführt, nicht jedoch für die Einrichtung von Untervezeichnissen.

*Beispiel*

In der Zentrale einer Lebensmittelkette steht ein Mainframe mit dem Betriebssystem BS2000 oder z/OS. In der Filiale stehen Windows- oder Unix-Rechner. Jeden Samstag stellt der Filialleiter einen Auftrag zum Übertragen der Datei, in der die Einnahmen der Woche aufgelistet und vorverarbeitet sind. Diese Datei wird mittels openFT an den Rechner der Zentrale übertragen. Im Übertragungsauftrag ist in der Folgeverarbeitung festgelegt, dass bei erfolgreicher Übertragung die Datei im Mainframe ausgedruckt und anschließend im Filialrechner gelöscht wird.




Dateiübertragung mit Folgeverarbeitung

### 2.6.2 Programmschnittstellen

openFT bietet durch seine Programmschnittstelle weitgehende Automatisierungsmöglichkeiten. Sie können dadurch z.B. die openFT-Auftragserteilung und -Auftragsverwaltung automatisieren, eigene Benutzeroberflächen für openFT erstellen oder File Transfer-Funktionen in andere Anwendungen integrieren. Für Windows-Systeme wird zusätzlich zur JAVA- und C-Programmschnittstelle auch eine OCX-Schnittstelle angeboten.

### 2.6.3 openFT-Script-Schnittstelle auf Unix und Windows-Systemen

openFT-Script stellt eine Skript-Sprache in XML-Notation zur Verfügung, die folgende von der Kommando- oder C-Schnittstelle gewohnte openFT-Funktionen umfasst:

– asynchrone Dateiübertragung

– Dateiverzeichnisse im fernen System erstellen

– Dateien oder Dateiverzeichnisse im fernen System löschen

– Dateiverzeichnisse im fernen System auflisten

– Kommando-Skripte im fernen System ausführen

Alle openFT-Script-Funktionen können auch auf lokale Dateien bzw. Dateiverzeichnisse angewandt werden.

Zusätzlich bietet openFT-Script im Vergleich zu den o.g. Schnittstellen folgende Vorteile:

– Logisch voneinander abhängige Einzelaufträge lassen sich zu einem Auftrag zusammenfassen und einfach überwachen.

– Einzelaufträge können sequenziell oder parallel ausgeführt werden.

– openFT-Script ist wiederanlauffähig. Wurde ein openFT-Script-Auftrag bei einem bestimmten Einzelauftrag unterbrochen, dann wird der openFT-Script-Auftrag beim Wiederanlauf an dieser Stelle fortgesetzt.

– openFT-Script-Aufträge können im openFT Explorer über das Objektverzeichnis *Ftscript Aufträge* überwacht und abgebrochen werden.

– Für Fehlerfälle (z.B. Partner nicht erreichbar, Datei nicht vorhanden usw.) können alternative Aktionen definiert werden.

### 2.6.4 Jobvariablen im BS2000

openFT (BS2000) bietet die Möglichkeit, einen Übertragungsauftrag durch eine Jobvariable überwachen zu lassen. Der Name der Jobvariablen sowie ein gegebenenfalls zum Zugriff auf diese Variable notwendiges Kennwort werden beim Übertragungsauftrag angegeben. Die Jobvariable, die einen Auftrag überwacht, dient auch zur Identifikation des Auftrages und kann als Selektionskriterium benutzt werden, um den Auftrag abzubrechen oder Informationen über den Auftrag zu erhalten. Zusätzlich kann sie zur Ereignissteuerung oder zur Änderung der Auftragsprioritäten herangezogen werden.

### 2.6.5 Automatisierte Weiterverarbeitung von openFT-Daten

Für die automatisierte Weiterverarbeitung von openFT-Daten (z.B. Logging-Sätze) durch openFT-fremde Prozeduren und Programme bietet openFT das sogenannte CSV-Ausgabeformat (**C**haracter **S**eparated **V**alues). Das CSV-Format steht für alle Anzeigekommandos von openFT zur Verfügung. Bei diesem Format werden die Informationen in je einer Textzeile ausgegeben, wobei die einzelnen Informationen eines "Ausgabe-Datensatzes" durch Strichpunkte getrennt werden. Die jeweils erste Zeile ist die Kopfzeile und enthält die Namen der Informationen, ebenfalls durch Strichpunkte getrennt.

Diese Ausgabe kann dann mit Programmen, die CSV-Formate unterstützen, weiterverarbeitet werden (z.B. unter Windows mit Microsoft Excel™). Damit lässt sich u.a. auf einfache Weise ein Accounting für verwendete Betriebsmittel wie z.B. Übertragungsaufträge realisieren.

Für z/OS bietet openFT auch einen Anschluss an SMF (System Mangement Facility), so dass openFT-spezifische Abrechnungssätze in die SFM-Datei geschrieben werden können.

## 2.7 Sicherer Betrieb

Offene Netze und Sicherheit bei der Datenübertragung und Dateimanagement sind Begriffe, die sich nicht widersprechen müssen. Die Funktionen, die Ihnen openFT für einen sicheren Betrieb bietet, sind:

– individuell einstellbarer Zugangs- und Zugriffsschutz mit der FTAC-Funktion
– Verschlüsselung bei der Dateiübertragung
– einstellbare Logging-Funktion
– Überprüfung des Kommunikationspartners durch Authentifizierung

Mit diesen Funktionen können Sie Ihr lokales System sicher machen.

### 2.7.1 Die FTAC-Funktion

Mit der FTAC-Funktion von openFT haben Sie alle Möglichkeiten in der Hand, Ihr System so sicher wie möglich und so sicher wie nötig zu machen. FTAC steht für "File Transfer Access Control".

FTAC bietet zum Schutz des eigenen Systems die folgenden Möglichkeiten:

– Entkopplung von FT-Zugangs- und Login-Berechtigung
– Zugriffsrechte abhängig von Partnersystemen
– benutzerspezifische Zugriffsrechte
– flexible Abstufung der Zugriffsrechte
– Protokollierung jeder Berechtigungsprüfung
– einfache Anwendung

#### 2.7.1.1 Leistungen der FTAC-Funktion

Bei der Übertragung von Dateien unterscheidet man verschiedene Funktionen. Für den Zugangs- und Zugriffsschutz ist dabei ausschlaggebend, was das zu schützende System bei der Dateiübertragung macht. Auf den ersten Blick gibt es nur zwei Funktionen:

– Senden einer Datei
– Empfangen einer Datei

Beim Senden einer Datei werden Daten aus dem zu schützenden System nach außen weitergegeben, beim Empfangen einer Datei gelangen Daten von außen in das zu schützende System. Nun besteht aber für den Datenschutz ein erheblicher Unterschied darin, wer von dem zu schützenden System eine Funktion verlangt. Im Sprachgebrauch des File Transfer heißt das, "wer Initiator (= Auftraggeber) eines Auftrages ist". Es sind zwei Gruppen von Auftraggebern zu unterscheiden:

– Auftraggeber im zu schützenden System (**Outbound-Aufträge**)
– Auftraggeber in Partnersystemen (**Inbound-Aufträge**)

Nach diesem Schema können jetzt die folgenden Funktionen unterschieden werden:

- **Outbound Senden**
- **Outbound Empfangen**
- **Inbound Senden**
- **Inbound Empfangen**

Als weitere Funktion muss beim File Transfer die Möglichkeit zur Verarbeitung der Übertragungsdaten (Vor-, Nach- und Folgeverarbeitung) beachtet werden. Für Aufträge, die im lokalen System selbst gestellt werden, erwächst daraus kein gesteigertes Schutzbedürfnis. Wer einen Auftrag im lokalen System erteilen darf, hat sowieso schon Zugriff auf die verfügbaren Betriebsmittel. Auch für Verarbeitungen, die in fernen Systemen ablaufen sollen, besteht kein Schutzbedürfnis im lokalen System. Eine Funktion, die Anforderungen an den Zugangsschutz im lokalen System stellt, ist die

- **Inbound Verarbeitung**

die von einem fernen System veranlasst wird.

Partnersysteme haben außerdem die Möglichkeit, mit den Dateimanagement-Funktionen sich in Ihrem lokalen System Dateiverzeichnisse oder Dateiattribute anzusehen, Dateiattribute zu ändern sowie Dateien und Verzeichnisse zu löschen. Daraus ergibt sich eine weitere Funktion:

- **Inbound Dateimanagement**

Das Dateimanagement umfasst im Gegensatz zu den anderen Funktionen mehrere verschiedenartige Auftragsmöglichkeiten, die wiederum teilweise mit den Funktionen *inbound senden* und *inbound empfangen* gekoppelt sind:

| **Inbound Dateimanagement-Funktion** | **Voraussetzung** |
|---|---|
| Dateiattribute anzeigen | Inbound Senden erlaubt |
| Dateiattribute ändern | Inbound Empfangen **und**<br>Inbound Dateimanagement erlaubt |
| Dateien umbenennen | Inbound Empfangen **und**<br>Inbound Dateimanagement erlaubt |
| Dateien löschen | Inbound Empfangen erlaubt |

Die Schutzmechanismen, die die FTAC-Funktion bietet, werden in erster Linie durch den Einsatz von Berechtigungssätzen und Berechtigungsprofilen erreicht.

### 2.7.1.2 Berechtigungssatz

Beim Einsatz von FTAC ist jeder Benutzerkennung im System ein individuell einstellbarer Berechtigungssatz zugeordnet. Außerdem gibt es den so genannten **Standard-Berechtigungssatz**, der vom FTAC-Verwalter festgelegt wird und der die Voreinstellungen und Vorgaben für alle individuellen Berechtigungssätze enthält.

Ein Berechtigungssatz enthält die grundsätzlichen Festlegungen für die File-Transfer-Funktionen. Für jeder der sechs File-Transfer Funktionen lassen sich Sicherheitsstufen zwischen 0 und 100 festlegen. 0 bedeutet Funktion gesperrt, 100 bedeutet Funktion uneingeschränkt erlaubt. Erfolgt ein Zugriff auf eine Benutzerkennung, so prüft FTAC, ob die im zugehörigen Berechtigungssatz eingestellten Sicherheitsstufen eingehalten werden.

### 2.7.1.3 Berechtigungsprofil

Mit einem Berechtigungsprofil definieren Sie die **FTAC-Zugangsberechtigung** und die damit verbundenen **Zugriffsrechte**. Die FTAC-Zugangsberechtigung (kurz Zugangsberechtigung) ist sozusagen der Schlüssel für den Zugriff via File Transfer auf Ihren Rechner. Deshalb sollten Sie die Zugangsberechtigung wie ein Kennwort behandeln. Sie muss bei Übertragungsaufträgen anstelle einer Login-Berechtigung angegeben werden. Eine Ausnahme bildet das Standard-Berechtigungsprofil einer Benutzerkennung, siehe Seite 47. Jeder, der diese Zugangsberechtigung kennt, hat zwar per File Transfer Zugang zu Ihrem Rechner, aber er kann im Gegensatz zur Login-Berechtigung nicht machen, was er will. Welche Funktionen Sie zulassen, legen Sie mit den Zugriffsrechten für diese Zugangsberechtigung fest. Sie regeln damit z.B.:

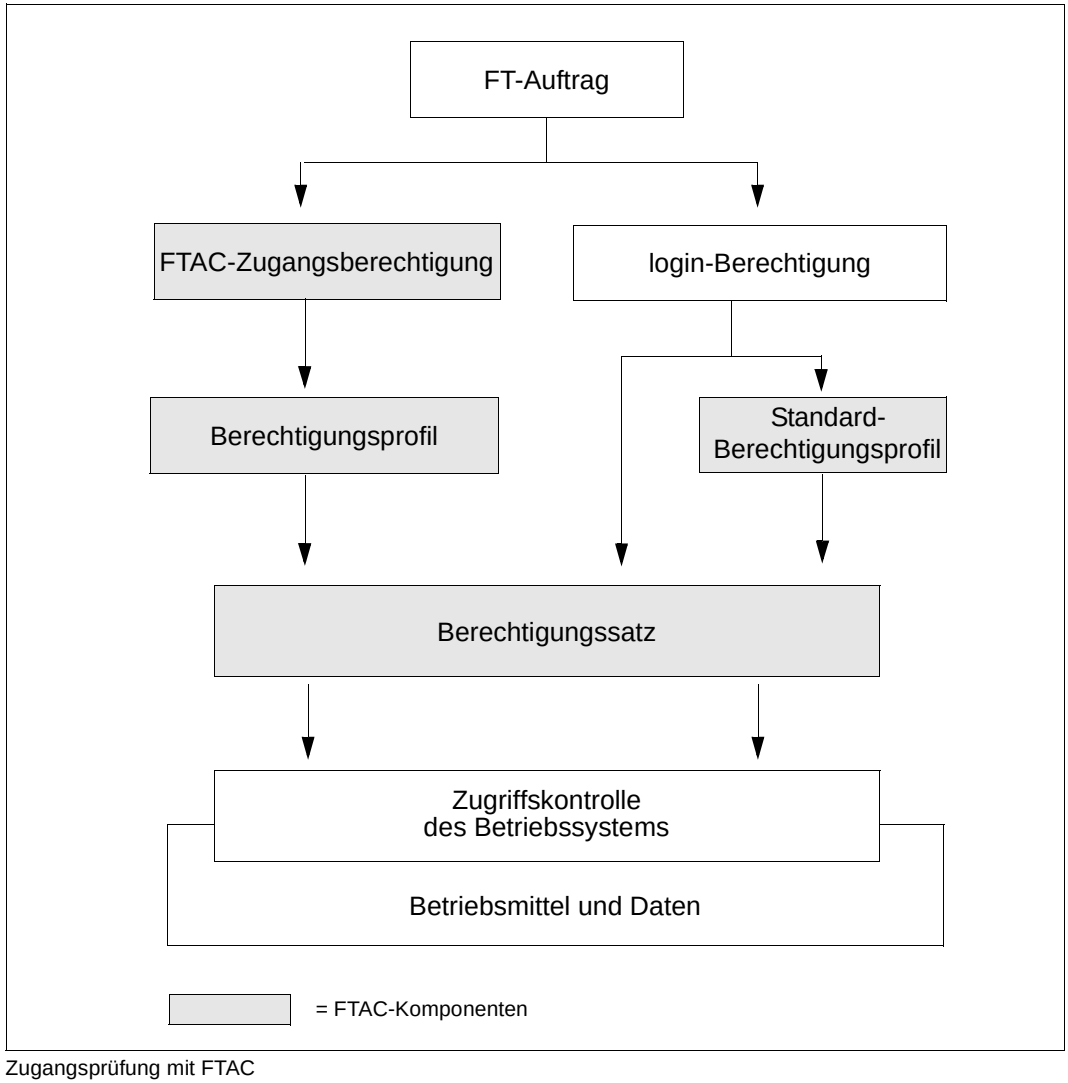
- auf welche Dateien unter welchen Voraussetzungen zugegriffen werden darf (Dateinamen-Präfix)
- welche Folgeverarbeitungskommandos nach der Übertragung erlaubt sind
- welche Partnersysteme auf dieses Berechtigungsprofil zugreifen dürfen

Im Extremfall können Sie so z.B. den Zugriff auf Ihren Rechner so einschränken, dass nur über ein einziges Profil auf nur eine Datei zugegriffen werden darf.

FTAC überprüft bei jedem File-Transfer-Auftrag, ob die Angaben im Auftrag im Widerspruch zu den Angaben im Berechtigungsprofil stehen. Ist das der Fall, wird der File-Transfer-Auftrag abgelehnt. Im fernen System erscheint dann nur eine allgemein gehaltene Fehlermeldung. Dadurch wird verhindert, dass jemand die Definitionen des Berechtigungsprofils durch schrittweises Ausprobieren ermitteln kann. Im lokalen System wird ein Logging-Satz erstellt, der die genaue Ursache beschreibt.

### Abläufe bei der Zugangsprüfung

Das folgende Bild zeigt die Abläufe bei der Zugangsprüfung mit FTAC.




Zugangsprüfung mit FTAC

**Standard-Berechtigungsprofil**

Für jede Benutzerkennung können Sie ein Standard-Berechtigungsprofil einrichten. Im Gegensatz zu einem normalen Profil besitzt ein Standard-Berechtigungsprofil keine FTAC-Zugangsberechtigung .

Dieses Profil ist für bestimmte Einsatzfälle gedacht, z.B. wenn ein FTAM-Partner für Inbound-Zugriffe die Zugangsberechtigung in einer festen Struktur angeben muss (Kennung und Passwort) und Sie dennoch bestimmte Einstellungen wie z.B. ein Dateinamen-Präfix vorgeben möchten.

## 2.7.2 Verschlüsselung bei Dateiübertragungsaufträgen

openFT bietet folgende Verschlüsselungsmöglichkeiten bei der Datenübertragung:

- Verschlüsslung der Auftragsbeschreibungsdaten:
  Damit sind die Protokolldaten gemeint, die der File Transfer zum Verbindungsaufbau und zur Auftragsbearbeitung verschickt und empfängt.
- Verschlüsselung der Benutzerdaten (d.h. Dateiinhalte, nur für openFT-Partner und Outbound Secure-FTP-Partner möglich):
  Die Verschlüsselung der Dateiinhalte kann individuell beim Übertragungsauftrag oder über ein Berechtigungsprofil oder generell per Betriebsparameter eingestellt werden. Dabei gibt es die Möglichkeit, die Verschlüsselung zu erzwingen oder zu verbieten, z.B. aus Performancegründen.

Bei Kopplung zu Partnern wird das Verschlüsselungsverfahren RSA/AES für die Auftragsbeschreibungsdaten und den Inhalt der zu übertragenden Datei verwendet.

Dazu benutzt openFT ab V12.0 standardmäßig einen 256 Bit AES-Schlüssel und einen 2048 Bit RSA-Schlüssel. Alternativ kann auch ein 1024 Bit oder 768 Bit RSA-Schlüssel verwendet werden.

Die Schlüssellänge muss der FT-Verwalter über die Betriebsparameter einstellen. Bei Kopplungen mit älteren Versionen wird die Verschlüsselung nötigenfalls heruntergehandelt, d.h. es wird ein RSA-Schlüssel mit einer dort verfügbaren Länge, oder, wenn dort kein AES-Schlüssel unterstützt wird, DES-Verschlüsselung eingesetzt.

Der FT-Verwalter kann auch eine Mindestschlüssellänge für den RSA- Schlüssel und/oder eine Mindestschlüssellänge für den AES-Schlüssel festlegen. In diesem Fall wird ein Auftrag abgelehnt, wenn der Partner diese Anforderung nicht erfüllen kann.

Der Mechanismus zur aktiven Verschlüsselung der Benutzerdaten ist aus rechtlichen Gründen eine eigene Liefereinheit und muss explizit freigeschaltet werden.

Weitere Details finden Sie im Abschnitt „Verschlüsselung bei der Datenübertragung“ auf Seite 131 und Abschnitt „Schutzmechanismen gegen Datenmanipulation“ auf Seite 133.

### 2.7.3 openFT-Betrieb protokollieren - die Logging-Funktion

Unerlaubte Zugriffe abzuwehren und Datenbestände zu schützen, ist nur ein Sicherheitsaspekt. Die lückenlose Dokumentation der Zugriffsüberprüfung und der Dateiübertragungsaufträge versetzt Sie darüberhinaus in die Lage, jederzeit Ihr Sicherheitsnetz überprüfen und ggf. undichte Stellen aufspüren zu können. Die Logging-Funktion von openFT ist das geeignete Werkzeug dafür. Sie ist standardmäßig eingeschaltet und protokolliert alle Informationen über Dateiübertragungsaufträge, unabhängig davon, ob die Initiative im lokalen oder fernen System liegt und ob die Übertragung erfolgreich war oder nicht. Dazu werden in die Logging-Datei entsprechende **Logging-Sätze** geschrieben. Der Umfang der Protokollierung ist einstellbar.

Die Logging-Funktion dient auch als Basis für das Erkennen von Einbruchsversuchen. Außerdem können durch die Logging-Funktion Daten zur Weiterverrechnung von Leistungen gewonnen werden (siehe auch Abschnitt „Automatisierte Weiterverarbeitung von openFT-Daten“ auf Seite 42).

#### Logging-Sätze

Wenn Ihr lokales System durch FTAC geschützt ist, überprüft zuerst FTAC alle Zugriffe auf Ihr System und protokolliert das Ergebnis in einem **FTAC-Logging-Satz**. Bei negativer Zugangsprüfung lehnt schon FTAC den Auftrag ab. Ist die Zugangsprüfung positiv, dann wird:

- bei einem Dateiübertragungsauftrag - falls der Auftrag zustandekommt - anschließend ein **FT-Logging-Satz** geschrieben, der protokolliert, ob der Auftrag erfolgreich ausgeführt oder warum er abgebrochen wurde. Damit können für einen Übertragungsauftrag zwei Logging-Sätze vorliegen.
- bei einem Fernadministrations-Auftrag ein **ADM-Logging-Satz** geschrieben, der protokolliert, ob der Auftrag erfolgreich ausgeführt oder warum er abgebrochen wurde.

Logging-Sätze, die Ihre Benutzerkennung betreffen, können Sie jederzeit ansehen. Logging-Sätze anderer Benutzerkennungen kann nur der jeweilige Verwalter ansehen.

Der Verwalter kann den Umfang der Protokollierung getrennt nach FT-, FTAC- und ADM-Logging sowie nach Verzeichnisübertragungs-Logging festlegen. Das FT- ADM- und Verzeichnisübertragungs-Logging kann komplett ausgeschaltet werden, das FTAC-Logging lässt sich ebenfalls einschränken, abgelehnte FTAC-Zugriffsüberprüfungen werden jedoch immer protokolliert.

*Offline-Logging*

Der FT-Verwalter kann die Logging-Datei im laufenden Betrieb wechseln. Nach dem Umschalten werden neue Logging-Sätze in eine neue Logging-Datei geschrieben. Die bisherige Logging-Datei bleibt als Offline-Logging-Datei bestehen. Sie können die Logging-Sätze Ihrer Benutzerkennung weiterhin mit openFT-Mitteln ansehen.

*Logging von Aufträgen mit Vor-/ Nachverarbeitung*

Aus Sicherheitsgründen werden nur die ersten 32 Zeichen (bzw. 42 Zeichen bei *ftexecsv* Vorverarbeitungen auf Unix- oder Windows-Systemen) eines Vor- oder Nachverarbeitungskommandos in den Logging-Satz übernommen.

**Logging-Sätze sichern und löschen**

Logging-Sätze und Logging-Dateien dürfen nur der FT-Verwalter, der FTAC-Verwalter und der ADM-Verwalter löschen. Logging-Sätze sollten in regelmäßigen Zeitabständen (am besten durch einen zyklischen Job) gesichert werden. Dabei sollte die Ausgabe des entsprechenden Kommandos gesichert werden, nicht die aktive Logging-Datei selbst. Durch Umschalten der Logging-Datei können die aktuellen Logging-Sätze in einer Offline-Logging-Datei gespeichert werden. Diese Offline-Logging-Datei können Sie als FT-Verwalter sichern. Dadurch bleiben zum einen die Logging-Sätze für eine lückenlose Dokumentation über einen längeren Zeitraum erhalten, zum anderen wird die Logging-Datei aber nicht größer als notwendig, was auch Rechenzeit beim Zugriff spart.

## 2.7.4 Authentifizierung

Sollen sicherheitskritische Daten übertragen werden, so ist es wichtig, das jeweilige Partnersystem vor der Übertragung einer sicheren Identitätsprüfung zu unterziehen ("Authentifizierung"). Die beiden an einer Übertragung beteiligten openFT-Instanzen können gegenseitig mit kryptografischen Mitteln überprüfen, ob sie mit der "richtigen" Partnerinstanz verbunden sind.

openFT unterstützt dazu folgendes Adressierungs- und Authentifizierungskonzept:

- Adressierung der openFT-Instanzen durch netzweit eindeutige Instanz-Identifikationen

  Für das lokale System wird diese Identifikation über einen Betriebsparameter definiert. Instanzidentifikationen von Partnersystemen werden in der Partnerliste hinterlegt. Anhand der Instanzidentifikationen der Partnersysteme verwaltet openFT die diesen Partnern zugeordneten Betriebsmittel wie z.B. Auftragswarteschlangen und kryptografische Schlüssel.

- Austausch von partnerspezifischen Schlüsselinformationen.

  Der FT-Verwalter kann für jede lokale openFT-Instanz RSA-Schlüsselpaarsätze bereitstellen, die jeweils aus einem privaten und einem öffentlichen Schlüssel bestehen.

  Damit die eigene openFT-Instanz im Partnersystem authentifiziert werden kann, muss dem Partnersystem der passende öffentliche Schlüssel zur Verfügung gestellt werden. Das sollte auf gesichertem Weg geschehen.

Weitere Details finden Sie in Abschnitt „Authentifizierung“ auf Seite 124.

## 2.8 openFT im Cluster oder HIPLEX-/SYSPLEX-Verbund einsetzen

In openFT gibt es die Möglichkeit, mehrere openFT-Instanzen auf einem Rechner gleichzeitig laufen zu lassen. Damit ist man in der Lage, bei einem Rechnerausfall die Funktionalität des openFT auf einen anderen Rechner im Cluster umzuschalten, auf dem bereits ein openFT läuft.

openFT-Kommandos, die in einer Vor-, Nach- oder Folgeverarbeitung aufgerufen werden, laufen unter der gleichen Instanz ab, wie der Auftrag, der die Vor-, Nach- oder Folgeverarbeitung initiiert hat.

# 3 Partnerkonzept

Das ferne System wird als Partnersystem oder kurz auch als Partner bezeichnet. Das Partnerkonzept von openFT beruht auf der so genannten Partnerliste.

Zusätzlich können Sie auf Unix- und Windows-Systemen, auf denen PCMX installiert ist den TNS nutzen, Abschnitt „Einsatz von PCMX und TNS in Unix- und Windows-Systemen“ auf Seite 58.

## 3.1 Partnerliste

Die Partnerliste wird vom FT-Verwalter eingerichtet und gepflegt.

Ein Partner in der Partnerliste wird durch seine Adresse definiert, siehe Abschnitt Partner-adressen. Zusätzlich können folgende Eigenschaften vergeben werden:

- Partnername
- Priorität
- Instanzidentifikation des Partners (wenn von Standard abweichend)
- Sicherheitseinstellungen wie z.B. FTAC-Sicherheitsstufen, siehe Seite 55
- Serialisierungseinstellungen, siehe Seite 56
- Inbound-/Outbound-Deaktivierung, siehe Seite 56
- Trace-Optionen (Ein/Aus)

Je nachdem, ob und in welcher Form ein Partnersystem in die Partnerliste eingetragen wird, unterscheidet man benannte Partner, eingetragene dynamische Partner und freie dynamische Partner, siehe Abschnitt „Partnertypen“ auf Seite 52.

### 3.1.1 Partneradressen

Eine Partneradresse hat bei openFT folgenden Aufbau:

[protocol://]host[:[port number].[T Selector].[Session Selector].[Presentation Selector]]

*host* (= Rechnername)
muss immer angegeben werden, alle anderen Angaben sind optional.

*protocol://*
Protokollstack, über den der Partner angesprochen wird.

Mögliche Werte für *protocol* sind:

- **openFT** (openFT-Protokoll, Standardwert)
- **ftam** (FTAM-Protokoll)
- **ftp** (FTP-Protokoll)
- **ftadm** (FTADM-Protokoll für Fernadministrationsaufträge)

*port number*, *T Selector, Session Selector*, *Presentation Selector*
werden nur in Sonderfällen oder bei FTAM-Partnern benötigt.

Weitere Details finden Sie in den openFT-Kommandohandbüchern.

### 3.1.2 Partnertypen

openFT kennt drei Partnertypen:

- Benannte Partner: Alle Partner, die mit Adresse und Namen in der Partnerliste eingetragen sind.
- Eingetragene dynamische Partner: Alle Partner, die mit Adresse, aber ohne Namen in der Partnerliste eingetragen sind.
- Freie dynamische Partner: Alle Partner, die nicht in der Partnerliste eingetragen sind.

Eingetragene dynamische Partner und freie dynamische Partner werden kurz auch als dynamische Partner bezeichnet.

#### Benannte Partner

Benannte Partner werden in FT-Aufträgen über ihren in der Partnerliste definierten Partnernamen angesprochen.

Diese Partner bleiben so lange in der Partnerliste, bis sie durch den FT-Verwalter gelöscht werden. Wenn eine Verbindung zu einem Partner mit Authentifizierung arbeiten soll, dann sollte dieser Partner in der Partnerliste eingetragen sein.

Die Verwendung von benannten Partnern bietet folgende Vorteile:

– Kompliziertere Partneradressen müssen in openFT-Aufträgen nicht explizit angegeben werden.
– Die Sicherheit wird erhöht, da nur wirklich bekannte Partner zugelassen werden können.
– openFT-Aufträge sind unabhängig von Adress-Änderungen: Prozeduren, Skripte oder Programme mit automatisierten openFT-Aufträgen verwenden den Partnernamen und müssen bei Adress-Änderungen nicht geändert werden.

Ein benannter Partner kann zwar auch über seine Adresse angesprochen werden, aber in allen Ausgaben von openFT wie z.B. beim Logging oder Auftragsbuch wird der Partnername angezeigt.

### Eingetragene dynamische Partner

Eingetragene dynamische Partner sind alle Partner, die nur mit Adresse, aber ohne Namen in der Partnerliste eingetragen sind. Sie können ausschließlich über ihre Adresse angesprochen werden und besitzen mindestens ein Attribut, das vom Standard eines freien dynamischen Partners abweicht, z.B. Trace einschalten (siehe Abschnitt "Freie dynamische Partner").

Wenn der FT-Verwalter für einen Partner dieses Typs alle Attribute auf die Standardwerte zurücksetzt, dann verschwindet dieser Partner aus der Partnerliste und wird zu einem freien dynamischen Partner.

### Freie dynamische Partner

Freie dynamische Partner sind alle Partner, die nicht in der Partnerliste eingetragen sind.

Partner dieses Typs werden nur über ihre Adresse angesprochen und besitzen mit Ausnahme der Sicherheitsstufe (siehe Abschnitt „FTAC-Sicherheitsstufen für Partner in der Partnerliste“ auf Seite 55) die Standard-Attribute, so wie sie beim Aufnehmen eines Partners in die Partnerliste gelten.

Das Konzept der freien dynamischen Partner bietet den Vorteil, dass ein Benutzer beliebige Partner adressieren kann, die nicht in der Partnerliste eingetragen sind. Als Administrator haben Sie dadurch weniger Verwaltungsaufwand. Nachteilig ist das erhöhte Sicherheitsrisiko, weshalb Sie die Verwendung dynamischer Partner auch untersagen können.

Sie können aus einem freien dynamischen Partner wie folgt einen eingetragenen dynamischen Partner machen:

- Geben Sie eine Partneradresse an, die auf keinen bereits existierenden Partnerlisteneintrag verweist.
- Legen Sie ein oder mehrere vom Standard abweichende Attribute (siehe oben) fest.

Wenn sich der Zustand eines freien dynamischen Partners ändert, (z.B. in NOCON = Partner nicht verfügbar) und damit vom Standardwert abweicht, wird er in der Partnerliste angezeigt. Er wird aber wieder zum freien dynamischen Partner, sobald er wieder verfügbar ist (Status ACTIVE).

### Dynamische Partner ein-/ausschalten

Per Betriebsparameter können Sie als FT-Verwalter die Verwendung dynamischer Partner aus Sicherheitsgründen untersagen. In diesem Fall muss ein Partner über seinen Namen in der Partnerliste adressiert werden; er kann nicht mehr direkt über seine Adresse angesprochen werden. Auch Inbound werden dann nur noch Partner zugelassen, die mit einem Partnernamen in der Partnerliste eingetragen sind.

## 3.2 FTAC-Sicherheitsstufen für Partner in der Partnerliste

Für den Fall, dass die FTAC-Funktionalität genutzt werden soll, sollte der FT-Verwalter in Abstimmung mit dem FTAC-Verwalter für jeden Partner in der Partnerliste zusätzlich die für FTAC relevante Sicherheitsstufe definieren.

Die Sicherheitsstufen sind die Maßeinheit für das Schutzbedürfnis gegenüber dem Partnersystem. Großes Schutzbedürfnis führt zu einer großen Sicherheitsstufe, kleines Schutzbedürfnis zu einer kleinen Sicherheitsstufe. Beim ersten Einsatz von FTAC sollten die Sicherheitsstufen in Zehnerschritten vergeben werden. Dadurch wird die Möglichkeit offen gelassen, neu hinzukommende Partnersysteme flexibel in die bestehende Hierarchie einzubetten.

Sicherheitsstufen können Sie entweder explizit vergeben (Minimalwert 0, Maximalwert 100) oder Sie können den Automatismus "Partnerattribute" aktivieren, d.h.:

- Partner, die von openFT authentifiziert werden, erhalten die Sicherheitsstufe 10.
- Partner, die im Transportsystem bekannt sind, erhalten die Sicherheitsstufe 90.
- Partner, die nur über ihre IP-Adresse adressiert werden (z.B. FTP-Partner) erhalten Sicherheitsstufe 100.

Dieser Automatismus kann sowohl partnerspezifisch als auch global per Betriebsparameter aktiviert werden.

Dieser Automatismus gilt auch für alle Partner, die nicht in der Partnerliste eingetragen sind (freie dynamische Partner), unabhängig davon, was in den Betriebsparametern eingestellt ist.

Wurde beim Erzeugen eines Partners keine Sicherheitsstufe angegeben, so verwendet openFT die globale Einstellung in den Betriebsparametern. Dort kann auch eine feste Sicherheitsstufe als Standard eingestellt werden.

#### Wirkung der Sicherheitsstufe

Die Sicherheitsstufe eines Partnereintrags kommt zum Tragen, wenn ein Benutzer über diesen Partnereintrag einen Auftrag abwickeln will. Die Sicherheitsstufe des Partnereintrags wird von FTAC mit der im Berechtigungssatz des Benutzers vergebenen Sicherheitsstufe für die benötigte Funktion (zum Beispiel inbound Senden) verglichen. Ist die Sicherheitsstufe im Berechtigungssatz kleiner als die des Partnereintrags, so wird der Auftrag von FTAC abgelehnt. Wird für den Auftrag ein privilegiertes FTAC-Profil verwendet, so kann der Benutzer sich über die im Berechtigungssatz definierten Einschränkungen hinwegsetzen.

## 3.3 Outbound- und Inbound-Deaktivierung von benannten Partnern

Sie haben die Möglichkeit, benannte Partner gezielt für asynchrone Outbound-Aufträge oder für Inbound-Aufträge zu deaktivieren.

Für Outbound-Aufträge können Sie zusätzlich die automatische Deaktivierung einschalten, sodass nach fünf fehlgeschlagenen Verbindungsaufbauversuchen der Partner für Outbound-Aufträge deaktiviert wird. Vor einem erneuten Versuch zum Verbindungsaufbau muss dieses Partnersystem manuell wieder aktiviert werden. Dadurch wird verhindert, dass unnötige Kosten entstehen, da u.U. auch erfolglose Verbindungsaufbauversuche kostenpflichtig sind.

## 3.4 Serialisierung von asynchronen Outbound-Aufträgen

Sie können für ein Partnersystem die Serialisierung von asynchronen Outbound-Aufträgen erzwingen.

Dadurch werden Überholeffekte vermieden, die bei paralleler Bearbeitung von Aufträgen auftreten können. Im Einzelnen gilt für die serielle Bearbeitung:

- Ein Folgeauftrag wird erst gestartet, wenn der vorhergehende Auftrag beendet ist.
- Die Serialisierung schließt Vor- und Nachverarbeitungen mit ein, nicht aber Folgeverarbeitungen, da diese unabhängig vom Auftrag sind.

Diese Funktion kann z.B. in einer Filial-Zentral-Konfiguration eingesetzt werden, bei der die Filialen zeitgleich mehrere Dateien an die Zentrale schicken (Tages-, Wochen- oder Monatsabschluss). Wird in den Filialrechnern für den Partner „Zentralrechner“ die Serialisierung aktiviert, dann kann pro Filialrechner immer nur eine Verbindung zum Zentralrechner aktiv sein. Dadurch werden Engpässe auf dem Zentralrechner verhindert wie z.B. eine regelmäßige Überschreitung des Verbindungslimits (Betriebsparameter CONN-LIM).

## 3.5 Priorisierung von Partnern

Partner können in der Partnerliste priorisiert werden. Diese Priorität kommt aber nur unter Aufträgen zum Tragen, die dieselbe Auftragspriorität haben, aber zu Partnern mit unterschiedlicher Partnerpriorität gehen. Ansonsten hat die Auftragspriorität Vorrang vor der Partnerpriorität.

Die folgenden Listen zeigen für die jeweiligen Plattformen, in welcher Reihenfolge Aufträge bearbeitet werden, wenn Aufträge mit verschiedenen Auftrags- und Partnerprioritäten vorhanden sind:

BS2000 und z/OS:

| **Bearbeitungs-reihenfolge** | **Auftragspriorität** | **Partnerpriorität** |
|---|---|---|
| 1 | hoch | hoch |
| 2 | hoch | normal |
| 3 | hoch | niedrig |
| 4 | normal | hoch |
| 5 | normal | normal |
| 6 | normal | niedrig |
| 7 | niedrig | hoch |
| 8 | niedrig | normal |
| 9 | niedrig | niedrig |

Unix- und Windows-Systeme:

| **Bearbeitungs-reihenfolge** | **Auftragspriorität** | **Partnerpriorität** |
|---|---|---|
| 1 | normal | hoch |
| 2 | normal | normal |
| 3 | normal | niedrig |
| 4 | niedrig | hoch |
| 5 | niedrig | normal |
| 6 | niedrig | niedrig |

## 3.6 Einsatz von PCMX und TNS in Unix- und Windows-Systemen

Ab openFT V10 ist die Verwendung des Transport Name Service (TNS) für Kopplung über TCP/IP nicht mehr notwendig. Wenn Sie den TNS dennoch nutzen, z.B. weil Sie mit anderen Transportsystemen als TCP/IP koppeln oder vorhandene TNS-Einträge nutzen möchten, muss PCMX installiert sein und der Betrieb mit PCMX und TNS muss explizit per Betriebsparameter eingeschaltet sein.

Der TNS identifiziert eine Transportsystem-Anwendung (TS-Anwendung) über einen symbolischen Namen, den sogenannten GLOBALEN NAMEN. Diesen symbolischen Namen werden Adressinformationen zugeordnet. Bei einem FT-Auftrag wird der Partner daher über den GLOBALEN NAME adressiert.

Beachten Sie bitte, dass die zusätzlichen Funktionen der Partnerliste wie z.B. FTAC-Sicherheitsstufen oder Inbound-/Outbound-Aktivierung für Partner nicht genutzt werden können.

**Wichtig**!
PCMX wird auf Unix-Systemen letztmalig mit openFT V12.1 unterstützt und ausgeliefert. Bitte stellen Sie alle TNS-Einträge, die openFT Partner betreffen, auf Partnerlisten-Einträge um. Dazu steht auf Unix- und Windows-Systemen die Prozedur *tns2ptn* zur Verfügung.

# 4 Dateiübertragung und Dateimanagement

Die Dateiübertragung mit openFT wird durch einen Dateiübertragungsauftrag initiiert. Im Dateiübertragungsauftrag machen Sie u.a. Angaben über den Partnerrechner, die Übertragungsrichtung, Dateinamen und Dateieigenschaften. Angesichts der Vielzahl unterstützter Hardware- und Software-Plattformen unterliegen diese Angaben den Konventionen der unterschiedlichen, am File Transfer beteiligten Betriebssysteme. Welche Dateien zwischen zwei Rechnern übertragen werden können, hängt davon ab, ob es sich bei den File Transfer-Partnern um gleichartige (homogene Kopplung) oder verschiedenartige Betriebssysteme (heterogene Kopplung) handelt. Ist am File Transfer ein Partner beteiligt, der die FTAM-Funktionalität nutzt, handelt es sich grundsätzlich um eine heterogene Kopplung. Das von openFT angebotene Dateimanagement gestattet es Ihnen, vor oder nach der Dateiübertragung bzw. auch ohne Dateiübertragung Dateien zu löschen, umzubenennen oder Dateiattribute zu ändern.

Die Nutzung der FTAC-Funktionalität bietet Ihnen nicht nur Sicherheitsvorteile, zusätzlich ermöglicht sie Ihnen, entsprechende FTAC-Einstellung auf den am File Transfer beteiligten Rechnern vorausgesetzt, Ihre Dateiübertragung betriebssystemunabhängig abzuwickeln (siehe Abschnitt „Die FTAC-Funktion“ auf Seite 43).

### Angaben beim Dateiübertragungsauftrag

Die folgenden Abschnitte geben Ihnen einen Überblick über die für einen Dateiübertragungsauftrag notwendigen Angaben. Die Angaben gliedern sich in einen lokalen, einen fernen und einen optionalen Teil. Im lokalen Teil geben Sie den lokalen Dateinamen ggf. mit dem entsprechenden Verzeichnisnamen und Dateikennworten an. Im fernen Teil definieren Sie den fernen Dateinamen, den Partnerrechner und Ihren Zugang zu diesem Rechner (Benutzerkennung, ggf. Abrechnungsnummer und Passwort bzw. Zugangsberechtigung). Im optionalen Teil haben Sie die Möglichkeit, Übertragungsmodalitäten wie z.B. Dateiarten, Schreibregeln und Folgeverarbeitungsaufträge festzulegen.

## 4.1 Dateinamen

Die folgende Beschreibung gibt einen Überblick über die systemspezifischen Konventionen zur Eingabe der Dateinamen unabhängig davon, ob es sich um die Angabe des lokalen oder fernen Dateinamens handelt. Die Nutzung der FTAC-Funktionalität kann Ihnen bei entsprechender Definition des Berechtigungsprofils die Eingabe des Dateinamens ganz oder teilweise ersparen (siehe Abschnitt „Berechtigungsprofil“ auf Seite 45), d.h. die Teile des Dateinamens, die im FTAC-Berechtigungsprofil definiert sind, dürfen Sie im Dateiübertragungsauftrag nicht mehr angeben.

### 4.1.1 Eindeutige Empfangsdateinamen

Ein wichtiger Anwendungsfall der openFT-Produkte ist die Übertragung einer Datei zu einem Zielrechner mit automatischer Weiterverarbeitung der empfangenen Datei. Oft ist dabei die Empfangsdatei nur ein Zwischenprodukt der Verarbeitung. Um Konfliktfälle durch parallel laufende Aufträge zu verhindern, kann durch Angabe der Metazeichenkette %UNIQUE oder %unique im Empfangsdateinamen das Anlegen eines eindeutigen Dateinamens durch openFT verlangt werden. openFT ersetzt %UNIQUE oder %unique durch einen String, der bei jedem erneuten Aufruf wechselt. Die Länge des Strings ist plattformabhängig.

In Logging-Sätzen, Ergebnisprotokollen, in der Statusanzeige von Übertragungsaufträgen sowie bei Meldungen wird der Dateiname bereits mit den für %UNIQUE eingesetzten Werten angezeigt.

### 4.1.2 BS2000-Dateinamen

| Format bei BS2000 (DVS) | Bedeutung |
|---|---|
| :cat:$user.dateiname | :cat:<br>wahlfreie Angabe der Katalogkennung;<br>Zeichenvorrat auf A…Z und 0…9 eingeschränkt; max. 4 Zeichen; ist in Doppelpunkte einzuschließen;<br>voreingestellt ist die Katalogkennung, die der Benutzerkennung laut Eintrag im Benutzerkatalog zugeordnet ist. |
| | $user.<br>wahlfreie Angabe der Benutzerkennung;<br>Zeichenvorrat ist A…Z, 0…9, $, #, @;<br>max. 8 Zeichen; darf nicht mit einer Ziffer beginnen; $ und Punkt müssen angegeben werden;<br>voreingestellt ist die Benutzerkennung, unter der auf die Datei zugegriffen wird. |
| | dateiname<br>Dateiname kann durch Punkt in mehrere Teilnamen gegliedert sein: $name_1$[.$name_2$[...]]<br>$name_i$ enthält keinen Punkt und darf nicht mit Bindestrich beginnen oder enden;<br>Zeichenvorrat ist A…Z, 0…9, $, #, @.<br>Länge max. 41 Zeichen, darf nicht mit $ beginnen und muss mindestens ein Zeichen aus A…Z enthalten. |
| :cat:$user.gruppe(gen-nr) | :cat:<br>siehe oben<br>$user.<br>siehe oben<br>gruppe<br>Name einer Dateigenerationsgruppe<br>Zeichenvorrat siehe "dateiname"<br>Klammern müssen angegeben werden,<br>Länge max. 41 Zeichen.<br>(gen-nr)<br>(*abs)<br>absolute Generationsnummer (1..9999);<br>* und Klammern müssen angegeben werden.<br>(+/-rel)<br>relative Generationsnummer (0..99);<br>Vorzeichen und Klammern müssen angegeben werden. |

| Format bei BS2000 (DVS) | Bedeutung |
|---|---|
| :cat:$user.lib/typ/element | :cat:<br>siehe oben<br>$user.<br>siehe oben<br>lib<br>Bibliotheksname;<br>es gelten die Regeln für BS2000-DVS-Dateinamen.<br>typ<br>Elementtyp;<br>1 - 8 Zeichen langer alphanumerischer Name.<br>element<br>Elementname;<br>es gelten die Regeln für LMS-Elementnamen;<br>element ist max. 64 Zeichen lang, darf nicht mit $ beginnen und muss mindestens ein Zeichen aus A…Z enthalten. |

Bei Dateiübertragungen mittels POSIX-Kommando müssen lokale DVS-Dateinamen inklusive catid angegeben werden. Namen ohne catid werden als POSIX-Dateien interpretiert. Für die default-catid kann das Präfix "::" verwendet werden.

### POSIX-Dateinamen

In den lokalen und/oder BS2000-Operanden für den POSIX-Dateinamen muss der POSIX-Dateiname als C-String (Graphic-String) - also in Hochkommas eingeschlossen - angegeben werden. Dies ist notwendig, um die Groß- und Kleinschreibung in POSIX-Dateinamen zu unterstützen.

| Format bei BS2000 (POSIX) | Bedeutung |
|---|---|
| posix-filename | Zeichenfolge, die maximal 255 Zeichen lang ist. Besteht entweder aus einem oder zwei Punkten, oder aus alphanumerischen Zeichen und Sonderzeichen; Sonderzeichen sind mit dem Zeichen \ zu entwerten. Nicht erlaubt ist das Zeichen /.<br>Muss in Hochkommas eingeschlossen werden, wenn alternative Datentypen zulässig sind, Separatoren verwendet werden oder das erste Zeichen ? bzw. ! ist.<br>Einem posix-filename muss ein posix-pathname vorangestellt werden. |

| Format bei BS2000 (POSIX) | Bedeutung |
|---|---|
| posix-pathname | Eingabeformat: [.]/[$part_1$/.../$part_n$]<br>wobei $part_n$ ein posix-filename ist;<br>maximal 1023 Zeichen.<br>Muss in Hochkommas eingeschlossen werden, wenn alternative Datentypen zulässig sind, Separatoren verwendet werden oder das erste Zeichen ? bzw. ! ist.<br>Der posix-pathname muss mit / bzw. ./ beginnen, oder zumindest aus / oder ./ bestehen. |

### 4.1.3 Dateinamen in Unix-Systemen

Bis zu 512 Zeichen, es wird zwischen Groß- und Kleinschreibung unterschieden. Von der Verwendung folgender Zeichen im Dateinamen wird abgeraten:
- ? @ # $ ^ & * ( ) ‘ [ ] \ | ; " < >.

### 4.1.4 Windows-Dateinamen

Dateiname meint hier den kompletten Pfadnamen.

Bis zu 256 Zeichen, folgende Zeichen dürfen nicht verwendet werden:
| * ? " < >.

Für ferne Dateinamen dürfen keine Netzlaufwerke angegeben werden, weder beim Holen noch beim Senden. Stattdessen können Sie UNC-Namen angeben.

#### UNC-Namen

UNC-Namen (**U**niversal **N**aming **C**onvention) sind Adressen freigegebener Ressourcen in einem Rechnernetz. Sie haben das Format:

```
\\rechnername\sharename\pfad\datei
```

Für *rechnername* kann z.B. der Hostname oder die IP-Adresse angegeben werden:

```
\\host1\versand\kataloge\winterausgabe.pdf
```

oder

```
\\172.30.88.14\versand\kataloge\winterausgabe.pdf
```

### 4.1.5 z/OS-Dateinamen

| Format bei z/OS | Bedeutung |
|---|---|
| ':S:first-qual.dateiname'<br>oder<br>:S:dateiname | Angabe für PS-Dataset<br>:S:<br>wahlfreie Präfixangabe zur Kennzeichnung eines PS-Datasets<br>first-qual<br>"first level qualifier"<br>Angabe Benutzerkennung (max. 7 Zeichen, Zeichenvorrat ist A…Z, 0…9, $, #, @; darf nicht mit einer Ziffer beginnen) oder Alias-Name (max. 8 Zeichen)<br>dateiname<br>teilqualifizierter Dateiname<br>kann durch Punkt in mehrere Teilnamen gegliedert sein: $name_1$[.$name_2$[...]]<br>$name_i$ ist max. 8 Zeichen lang, Zeichenvorrat ist A…Z, 0…9,-, $, #, @; darf nicht mit einer Ziffer oder mit "-" beginnen<br>der teilqualifizierte Dateiname ist max. 36 Zeichen lang; der vollqualifizierte Dateiname (first-qual.dateiname) ist max. 44 Zeichen lang. |
| ':S:first-qual.gen-gruppe.<br>GmmmmVnn'<br>oder<br>:S:gen-gruppe.GmmmmVnn | Angabe für absolute Dateigeneration<br>:S:<br>wahlfreie Präfixangabe zur Kennzeichnung eines PS-Datasets<br>first-qual<br>Syntax siehe "Angabe für PS-Dataset"<br>gen-gruppe<br>Syntax siehe "dateiname" in "Angabe für PS-Dataset"<br>Ausnahme: teilqualifizierter Dateiname der Dateigenerationsgruppe max. 27 Zeichen<br>vollqualifizierter Dateiname max. 35 Zeichen<br>GmmmmVnn<br>absolute Dateigeneration<br>mmmm absolute Generationsnummer (0000 - 9999)<br>nn Versionsnummer (00 - 99) |

| Format bei z/OS | Bedeutung |
|---|---|
| ':S:first-qual.<br>gen-gruppe(rel-gen-nr)'<br>oder<br>:S:gen-gruppe(rel-gen-nr) | Angabe für relative Dateigeneration<br>:S:<br>wahlfreie Präfixangabe zur Kennzeichnung eines PS-Datasets<br>first-qual<br>Syntax siehe "Angabe für PS-Dataset"<br>gen-gruppe<br>Syntax siehe "gen-gruppe" in "Angabe für absolute Dateigeneration"<br>rel-gen-nr<br>relative Generationsnummer<br>0 = aktuelle Generation<br>+/-m = 1 - 99 bei teilqualifizierter Angabe (ohne first-qual und Hochkommas)<br>1 - 255 bei vollqualifizierter Angabe (mit first-qual und Hochkommas) |
| ':prefix:first-qual.<br>dateiname(membername)'<br>oder<br>:prefix:dateiname(membername) | Angabe für PO- oder PDSE-Member<br>:prefix:<br>wahlfreie Präfixangabe zur Kennzeichnung der Dateiorganisation;<br>kann folgende Werte annehmen:<br>:O: für PO<br>:E: für PDSE<br>:L: für PO oder PDSE<br>first-qual<br>Syntax siehe "Angabe für PS-Dataset"<br>dateiname<br>teilqualifizierter Dateiname des PO- oder PDSE-Datasets<br>Syntax siehe "dateiname" in "Angabe für PS-Dataset"<br>membername<br>Name des PO- oder PDSE-Members<br>max. 8 Zeichen lang, Zeichenvorrat ist A…Z, 0…9, $, #, @;<br>darf nicht mit einer Ziffer beginnen |
| ':V:first-qual.dateiname'<br>oder<br>:V:dateiname | Angabe für VSAM-Datei vom Typ "entry-sequenced"<br>:V:<br>wahlfreie Präfixangabe zur Kennzeichnung einer VSAM-Datei vom Typ "entry-sequenced"<br>first-qual<br>Syntax siehe "Angabe für PS-Dataset"<br>dateiname<br>teilqualifizierter Dateiname der VSAM-Datei<br>Syntax siehe "dateiname" in "Angabe für PS-Dataset" |

| Format bei z/OS | Bedeutung |
|---|---|
| ':prefix:first-qual.dateiname'<br>oder<br>:prefix:dateiname | Angabe für ein vollständiges PO- oder PDSE-Dataset<br>:prefix:<br>wahlfreie Präfixangabe zur Kennzeichnung der Dateiorganisation;<br>kann folgende Werte annehmen:<br>:O: für PO<br>:E: für PDSE<br>:L: für PO oder PDSE<br>first-qual<br>Syntax siehe "Angabe für PS-Dataset"<br>dateiname<br>teilqualifizierter Dateiname des PO- oder PDSE-Datasets<br>Syntax siehe "dateiname" in "Angabe für PS-Dataset"<br>Ausnahme: max. Länge des teilqualifizierten Dateinamens beträgt 34 Zeichen, vollqualifizierter Dateiname 42 Zeichen; d.h.die maximal mögliche Dateinamenslänge ist sowohl bei teil- als auch bei vollqualifizierter Angabe jeweils um 2 Zeichen kürzer als für ein PS-Dataset, da der Name des beim Transfer eines vollständigen PO- oder PDSE-Datasets benötigten temporären Datasets durch Anhängen von ".U" gebildet wird, siehe Handbuch "openFT (z/OS) - Kommandoschnittstelle". |

### openEdition-Dateien

Die Dateinamen entsprechen den POSIX-Konventionen.

| Format bei z/OS | Bedeutung |
|---|---|
| filename | Komponente für einen openEdition-Dateinamen.<br>Zeichenfolge, die maximal 255 Zeichen lang ist. Besteht entweder aus einem oder zwei Punkten, oder aus alphanumerischen Zeichen und Sonderzeichen. Nicht erlaubt ist das Zeichen /. |
| pathname | openEdition-Dateiname<br>Eingabeformat: [.]/[$part_1$/.../$part_n$]<br>wobei $part_n$ ein posix-filename ist;<br>maximal 512 Zeichen.<br>Beginnt der Name mit /, so wird er als absoluter Pfadname verstanden.<br>Beginnt der Name mit ./, so entspricht dies einem „relativen“ Pfadnamen, bezogen auf das Verzeichnis der Benutzerkennung, z.B. /u/<benutzerkennung in Kleinbuchstaben>/. |

## 4.2 Dateikennworte

Ist eine Datei, auf die mit openFT zugegriffen werden soll, mit Dateikennworten versehen, müssen diese angegeben werden. In Windows- und Unix-Systemen existieren keine Dateikennworte.

| System | Dateikennwort |
|---|---|
| BS2000 | 1 - 4 Zeichen langer C-String (Graphic-String) bzw.<br>1 - 8 Zeichen langer X-String (Octet-String) bzw.<br>Integer-String zwischen - 2147483648 und 2147483647 |
| z/OS | 1 - 8 alphanumerische Zeichen |

## 4.3 Dateiarten

Die Dateien, die übertragen werden können, haben abhängig von ihrer Dateiart und vom Betriebssystem, aus dem sie kommen, unterschiedliche Eigenschaften, die bei der Übertragung beachtet werden müssen.

### 4.3.1 BS2000-Dateien

Entsprechend der unterschiedlichen Dateistrukturen wird zwischen folgenden BS2000-Dateiarten unterschieden:

- Katalogisierte Dateien
  - DVS-Dateien (dazu zählen SAM-, ISAM-, PAM-Dateien, PLAM-Bibliotheken und katalogisierte Generationen einer Dateigenerationsgruppe)
  - POSIX-Dateien.
- Elemente einer katalogisierten PLAM-Bibliothek
  - abdruckbare bzw. benutzerdefinierte Elemente vom Typ D, J, M, S und ggf. X
  - Elemente mit BS2000-spezifischem Binärcode vom Typ C, L, R und ggf. X

Um POSIX-Dateien mit openFT übertragen zu können, muss POSIX gestartet sein. Das POSIX-Dateisystem entspricht von Aufbau und Struktur her im wesentlichen dem Unix-Dateisystem. Die folgende Übersicht zeigt den Zusammenhang zwischen Dateinamenssyntax und Dateiart im BS2000.

| **Dateinamens-Syntax** | **Dateiart** |
|---|---|
| beginnt mit $userid oder :catid:$userid und enthält keinen '/ ' | DVS-Datei, vollqualifiziert |
| beginnt weder mit '/' noch mit './' noch mit $userid noch mit :catid:$userid und enthält keinen '/' | DVS-Datei,<br>Pfad relativ zur Zugangsberechtigung |
| beginnt mit '/' | POSIX-Datei, vollqualifiziert |
| beginnt mit './' | POSIX-Datei, Pfad relativ zur Zugangsberechtigung |
| beginnt mit $userid oder :catid:$userid und enthält mindestens einen '/' | Name eines PLAM-Elements, vollqualifiziert |
| beginnt weder mit '/' noch mit './' noch mit $userid noch mit :catid:$userid aber enthält mindestens einen '/' | Name eines PLAM-Elements,<br>Pfad relativ zur Zugangsberechtigung |

BS2000-Dateien können sich sowohl auf gemeinschaftlichen als auch auf privaten Datenträgern (Platten) befinden. Für die Bearbeitung von Dateien auf privaten Datenträgern ist Voraussetzung, dass die Dateien katalogisiert und die privaten Datenträger ordnungsgemäß an das System angeschlossen sind.

#### Übertragen von Bibliothekselementen

FT-Systeme übertragen mit einem FT-Auftrag genau ein Bibliothekselement. Bei der Übertragung von Bibliothekselementen untereinander bleibt die Bibliotheksorganisation, die Zugriffsmethode sowie die Satzstruktur erhalten, während bei Übertragungen zwischen Bibliothekselementen und SAM-Dateien nur die Satzstruktur erhalten bleibt.

– Beim File Transfer mit FTAM-Partnern kann auf Bibliothekselemente, die im fernen Rechner vorliegen, nur per C-String (und nicht über *LIB-ELEM()) zugegriffen werden.

– Bibliothekselemente eines bestimmten Typs (z.B. Lademodule) können nur dann übertragen werden, wenn das FT-Produkt des Partnersystems das Übertragen dieses Elementtyps zulässt.

### 4.3.2 z/OS-Dateien

openFT (z/OS) kann folgende Arten von Dateien übertragen:

– PS-Datasets einschließlich absoluter und relativer Dateigenerationen
– Member von PO- und PDSE-Datasets (mit Ausnahme von Objektmodulen und Programmen)
– VSAM-Dateien vom Typ "entry-sequenced"
– openEdition-Dateien (Dateien der z/OS Unix Systems Services)
– Migrierte Dateien, d.h. mit HSM ausgelagerte Dateien, siehe auch Abschnitt „Ausgelagerte Dateien“ auf Seite 80.

Die Übertragung dieser Dateien erfolgt sequenziell. Die Dateien können homogen zwischen zwei z/OS-Systemen oder heterogen mit einem "Nicht"-z/OS-System übertragen werden. Beim homogenen File Transfer können alle Dateiarten aufeinander abgebildet werden. Zwischen z/OS und anderen Plattformen (heterogene Kopplung) ist eine Übertragung von Dateien möglich, wenn das fremde System ebenfalls sequenzielle Dateien unterstützt. Mit BS2000-Systemen können z.B. SAM-Dateien und PLAM-Elemente entsprechenden Typs ausgetauscht werden.

Die Übertragung kompletter PO- und PDSE-Datasets kann nur zwischen zwei z/OS-Systemen stattfinden.

z/OS-Dateien können sich sowohl auf gemeinschaftlichen als auch auf privaten Datenträgern (Platten) befinden. Für die Bearbeitung von Dateien auf privaten Datenträgern ist Voraussetzung, dass die Dateien katalogisiert und die privaten Datenträger ordnungsgemäß an das System angeschlossen sind.

Die Übertragung von Dateien mit dem Attribut "unmovable" (data organisation PSU) ist mit openFT nicht möglich.

### 4.3.3 Unix- und Windows-Dateien

Dateien in Unix- bzw. Windows-Systemen haben ebenso wie die POSIX-Dateien im BS2000-System keine Struktur und auch keine Dateimerkmale, die über die Codierung Auskunft geben. Windows-Dateien bieten, obwohl auch sie keine Struktur haben, Unterscheidungsmöglichkeiten anhand ihrer Dateinamens-Extension (z.B. "txt" für Text- und "exe" für ausführbare Dateien).

Für die Übertragung mit Windows- oder Unix-Systemen können Sie daher folgende Dateiarten definieren:

– Text
– unstrukturierte Binärdaten
– satzstrukturierte Binärdaten (Benutzerformat)

#### Textformat

Eine Datei, die von Windows- oder Unix-Systemen aus im Textformat versendet wird, muss eine reine Textdatei sein, die durch Zeilenende-Kennzeichen (Linefeed in Unix-Systemen bzw. Carriage Return und Linefeed in Windows) in Sätze strukturiert ist. Die Länge einer Zeile ist beschränkt, z.B. auf 98304 Bytes bei Windows-Systemen. Für die Übertragung wird das Zeilenende-Kennzeichen in jeder Zeile eliminiert.

Beim Übertragen von BS2000-Systemen oder z/OS-Systemen in Windows- oder Unix-Systeme wird das Zeilenende-Kennzeichen entsprechend den im fernen System bestehenden Satzlängen eingefügt. Der Text und die Satzlängen bleiben erhalten. Die maximale Satzlänge bei der Übertragung einer Textdatei ist betriebssystemabhängig.

Mit Partnersystemen ab openFT V10 können auch Unicode-Dateien übertragen werden, siehe Abschnitt „Übertragung von 7 Bit-, 8 Bit- und Unicode-Dateien“ auf Seite 81.

*Tabulator- und Leerzeilenexpansion*

Beim Übertragen von Textdateien führt openFT ggf. eine Tabulator- und Leerzeilenexpansion durch. D.h. anstelle eines Tabulators werden Leerzeichen und anstelle einer Leerzeile wird eine Zeile mit einem Leerzeichen übertragen. Dabei werden für openFT-Partner folgende Fälle unterschieden:

| Initiator | Richtung | Responder | Expansion (ja/nein) |
|---|---|---|---|
| Unix-, Windows-System | Senden | Unix-, Windows-System | nein, optional ja[1] |
| Unix-, Windows-System | Holen | Unix-, Windows-System | nein |
| Unix-, Windows-System | Senden | BS2000, z/OS | ja, optional nein[1] |
| Unix-, Windows-System | Holen | BS2000, z/OS | nein (nicht relevant) |
| BS2000, z/OS | Senden | Unix-, Windows-System | nein (nicht relevant) |
| BS2000, z/OS | Holen | Unix-, Windows-System | ja (beim Initiator) |
| BS2000, z/OS | Senden und Holen | BS2000, z/OS | nein |

[1] Die Expansion kann im Unix- oder Windows-System beim Auftrag explizit ein- bzw. ausgeschaltet werden.

Bei der Dateiübertragung mit FTAM-Partnern gibt es keine Leerzeilenexpansion. Bei Übertragungen mit dem Zeichensatz *GraphicString* werden Tabulatoren expandiert, im *GeneralString* hingegen nicht. Näheres zu FTAM-Zeichensätzen siehe auch Abschnitt „FTAM-Dateien“ auf Seite 73.

### Binärformat

Bei der Angabe "Binärformat" wird erwartet, dass die zu übertragende Datei eine unstrukturierte Folge von Binärdaten enthält. Im Empfangssystem wird eine Datei mit undefinierter Satzlänge (d.h. ohne Zeilenende-Kennzeichen) erzeugt. Die Binärdaten bleiben erhalten.

### Benutzerformat

Beim Senden von Dateien im Benutzerformat wird davon ausgegangen, dass die zu sendende Datei durch Längenfelder in Sätze strukturiert ist. Die ersten beiden Byte eines jeden Satzes müssen dessen Länge inklusive der Länge des Satzlängenfeldes enthalten. Beim Holen werden diese Längenangaben entsprechend der im fernen System bestehenden Satzlängen erzeugt. Der Inhalt der Sätze wird als Binärdaten behandelt, d.h. er wird nicht umcodiert.

Die Satzstruktur und die Binärdaten bleiben bei der Übertragung erhalten. Die Satzlängenfelder werden in Unix- und Windows-Systemen mit dem höchstwertigen Byte zuerst abgespeichert. Die maximale Satzlänge innerhalb einer Datei im Benutzerformat ist betriebssystemabhängig.

### 4.3.4 FTAM-Dateien

Mit FTAM-Partnern können Sie die sogenannten "document types" FTAM-1 (für Textdateien) und FTAM-3 (für Binärdateien) austauschen. Diese Funktion steht auf z/OS nicht zur Verfügung.

In der Kernel group in "contents-type" sind die Dateistruktur und Dateiinhalt dieser FTAM-Dateien beschrieben:

- constraint set
  beschreibt die Dateistruktur. Die durch die Funktionsnorm ISO/EN 10607-3 ausgewählte Untermenge der FTAM-Norm lässt nur den Wert *unstructured* (unstrukturiert) zu. *constraint set* legt außerdem aufgrund der Struktur der Datei fest, welche Aktionen mit dieser Datei erlaubt sind. Für unstrukturierte Dateien sind Lesen, Überschreiben, Erweitern und Löschen des Dateiinhalts erlaubt. Zusammen mit den *permitted actions* schränkt *constraint set* also die Menge der möglichen Aktionen ein.
- document type
  beschreibt den eigentlichen Inhalt der Datei. ISO/EN 10607-3 fordert die Unterstützung von FTAM-1 (unstructured text) für Text-Dateien und FTAM-3 (unstructured binary) für Dateien mit binärem Inhalt. Das String-Format (*string significance*) kann variabel (*variable*), fest (*fix*) oder ohne Bedeutung für die Abspeicherung (*not significant*) sein. Zusätzlich kann eine maximale Länge der Strings (*maximum string length*) definiert sein.

  Bei Textdateien (FTAM-1) wird mit der *universal class number* angegeben, welche Zeichen die Datei enthält:

  – *GraphicString* kann alle grafischen Zeichensätze (G sets) enthalten, wobei Escape-Sequenzen zwischen Zeichensätzen umschalten können (siehe ISO 2022).

    openFT setzt den Zeichenvorrat auf ISO 646 IRV (bzw. ASCII IRV oder ISO 8859-1 G0-Set) plus ISO 8859-1 G1-Set, der im Wesentlichen die Zeichen des europäischen Sprachraums abdeckt. Bei Kopplung zwischen zwei Partnern mit openFT ab V10 wird der Zeichenvorrat für die Dateiübertragung auf UTF-8 gesetzt.

  – *GeneralString* kann neben allen grafischen Zeichen auch Steuerzeichensätze (C sets) enthalten, die ebenfalls umschaltbar sind.

  – *VisibleString* enthält nur grafische Zeichen aus ISO 646 IRV.

  – *IA5String* enthält grafische Zeichen aus ISO 646 IRV und Steuerzeichen aus ISO 646 (C0 set).

### 4.3.5 Übertragung verschiedener Dateiarten

Ziel einer Dateiübertragung ist neben der vollständigen Übertragung des Dateiinhalts auch die authentische Darstellung der Dateistruktur. Werden wie bei homogenen Kopplungen üblich, gleiche Strukturen aufeinander abgebildet, ist die Authentizität problemlos zu realisieren, d.h. der Binärcode und die Zeichendarstellung sind im Sende- und Empfangssystem identisch. Bei heterogenen Kopplungen ist es meist nicht möglich, sowohl den Binärcode als auch die Zeichendarstellung im Empfangssystem unverändert zu erhalten. Daher wird bei der Dateiübertragung mit openFT zwischen Text- und Binärübertragung unterschieden. Spezifika zum File Transfer mit FTAM-Partnern entnehmen Sie bitte dem Abschnitt „Besonderheiten beim File Transfer mit FTAM-Partnern“ auf Seite 107.

#### Textübertragung

Die Textübertragung erfolgt zeichenorientiert, d.h. die Darstellung der Zeichen bleibt erhalten. Dies gilt sowohl für Zeichen im Einbyte-Code wie ISO 8859 als auch für Unicode-Zeichen, die in mehreren Bytes dargestellt werden. Die Satzstruktur der Textdatei wird im Empfangssystem nach den dort geltenden Konventionen systemkonform angepasst.

In den "Nutzdaten" einer Datei, die per Textübertragung transferiert wird, dürfen keine Zeichen verwendet werden, die vom jeweiligen System als Steuerzeichen interpretiert werden, z.B. X‘15‘ (EBCDIC-Linefeed) und X‘0A‘ (ASCII-Linefeed).

Bei der Textübertragung kann openFT (z/OS) dateispezifische Konvertiertabellen verwenden, die über den Dateinamen ausgewählt werden; bitte fragen Sie Ihren FT-Verwalter.

Die folgende Tabelle zeigt die möglichen Einstellungen für Textübertragung:

| **Satzstruktur im Empfangssystem** | **lokales System** | **fernes System** | **Richtung ← / →[1]** | **Datentyp/ DATA-TYPE** |
|---|---|---|---|---|
| systemkonform (in der im Empfangssystem üblichen Weise) | BS2000:DVS, PLAM | BS2000: DVS, PLAM | ← / → | Standard<br>Text<br>Binär<br>Benutzer |
| | BS2000: DVS, PLAM, POSIX | BS2000: POSIX, Unix-System, Windows, VMS, z/OS | ← / → | Standard<br>Text |
| | Unix- oder Windows-System | BS2000, z/OS | ← / → | Standard<br>Text |
| | Unix- oder Windows-System | Unix- oder Windows-System | ← / → | Standard<br>Text<br>Binär |

[1] ← = Holen, → = Senden

Details zu z/OS finden Sie im Handbuch "openFT (z/OS) - Kommandoschnittstelle".

### Binärübertragung

Die Binärübertragung erfolgt so, dass die Codierung (Binärdarstellung) der Zeichen erhalten bleibt. Die Gestaltung der Satzstruktur ist steuerbar. So passt openFT die Satzstruktur an die Satzstruktur des Empfangssystems an (systemkonforme Satzstruktur). Mit der originalen Satzstruktur bleibt die Struktur des Sendesystems erhalten. Außerdem besteht die Möglichkeit, mit dem FT-spezifischen Benutzerformat eigene, systemunabhängige Satzstrukturen zu verwenden.

Es ist nicht möglich, über das FTP-Protokoll Dateien mit Sätzen fester oder variabler Länge im Binärformat zu holen. Dies betrifft insbesondere auch die Ausgabe von Dateiübertragungen mit Vorverarbeitung auf BS2000 oder z/OS sowie von Ausgaben von per *ftexec* ausgeführten Kommandos auf BS2000 oder z/OS. Hier müssen Sie entweder die Dateien im Textformat übertragen oder ein anderes Übertragungsprotokoll (openFT) verwenden.

Die folgende Tabelle zeigt die möglichen Einstellungen für eine Binärübertragung:

| **Satzstruktur im Empfangssystem** | **lokales System** | **fernes System** | **Richtung ← / →[1]** | **Datentyp/ DATA-TYPE** |
|---|---|---|---|---|
| systemkonform (in der im Empfangssystem üblichen Weise) | BS2000: DVS, PLAM | DVS, PLAM, z/OS | ← / → | Standard<br>Text<br>Binär<br>Benutzer |
| | BS2000: DVS, PLAM | POSIX | ← / → | Standard<br>Text |
| | BS2000: POSIX | POSIX | ← / → | Standard<br>Text<br>Binär |
| | BS2000: POSIX | z/OS | ← | Text<br>Binär |
| | Unix- oder Windows-System | Unix-System, Windows | ← / → | Standard<br>Text<br>Binär |

| Satzstruktur im Empfangssystem | lokales System | fernes System | Richtung ← / →[1] | Datentyp/ DATA-TYPE |
|---|---|---|---|---|
| originale Satzstruktur (in der im Sendesystem üblichen Weise) | BS2000: POSIX | DVS, PLAM | → | Binär |
| | BS2000: POSIX | Unix-System, Windows-System | ← / → | Binär |
| | BS2000: DVS, PLAM | Unix-System, Windows-System | ← | Binär |
| | Unix- oder Windows-System | DVS, PLAM, z/OS | → | Binär |
| | Unix- oder Windows-System | POSIX, Windows, Unix-System, VMS | ← / → | Binär |
| Benutzerformat (systemunabängig) | BS2000: DVS, PLAM | POSIX | → | Benutzer |
| | BS2000: DVS[2], PLAM, POSIX | Unix-System, Windows-System | → | Benutzer |
| | BS2000: POSIX | Unix-System, Windows-System | ← | Benutzer |
| | Unix- oder Windows-System | DVS, PLAM, POSIX, z/OS | ←[3] | Benutzer |
| keine Satzstruktur (d.h. die Satzstruktur geht evtl. verloren) | BS2000: DVS, PLAM | POSIX, Unix-System, Windows, VMS | → | Binär |
| | Unix- oder Windows-System | DVS, PLAM, z/OS | ← | Binär |

[1] ← = Holen, → = Senden

[2] gilt nur für SAM-Dateien mit variabler Satzlänge (RECFORM=V)

[3] Sofern eine Datei im Benutzerformat geholt wurde, kann sie auch im Benutzerformat gesendet werden.

Temporäre BS2000-Dateien können mit openFT nicht übertragen werden.

ISAM- und PAM-Dateien können wie folgt zwischen BS2000- und Fremdsystemen übertragen werden:

- im Directory-Format, siehe Seite 78
- unter Angabe des Zielformates, siehe Abschnitt „Heterogene Übertragung von PAM- und ISAM-Dateien“ auf Seite 79

Details zu z/OS finden Sie im Handbuch "openFT (z/OS) - Kommandoschnittstelle".

### Satzweise Übertragung

*Lokales Unix- oder Windows-System*

Bei der Übertragung von Dateien zwischen Unix- oder Windows- und BS2000-Systemen kann die Satzstruktur von Bedeutung sein. Wenn Dateien aus Unix- oder Windows-Systemen in eine DVS-Datei übertragen werden, dann müssen Sie die maximale Satzlänge erhöhen, falls die als Standard erzeugten Blockgrößen der DVS-Dateien nicht ausreichen, um den längsten Satz aufzunehmen. Dies ist in der Regel ab einer Netto-Satzlänge von 2024-2040 Bytes der Fall.

*Lokales BS2000-System*

Bei der Übertragung von DVS-Dateien zwischen BS2000-Systemen wird im allgemeinen die Dateistruktur nicht beachtet (die Dateien werden blockweise übertragen). In folgenden Fällen ist die Satzstruktur von Dateien von Bedeutung (die Dateien werden satzweise übertragen):

– Übertragung zwischen BS2000 und Unix-, Windows- oder z/OS-Systemen
– Erweitern einer Datei mit Satzstruktur
– Übertragung von POSIX-Dateien
– Übertragung von Bibliothekselementen

Dabei darf die maximale Länge von zu übertragenden Sätzen folgende Werte nicht übersteigen:

- Partnersysteme mit openFT ab V10:
  - 32768 Byte bei Dateien mit fixer Satzlänge
  - 32760 Byte bei Dateien mit variabler Satzlänge
- Partnersysteme mit openFT < V10:
  - 32760 Byte bei Dateien mit fixer Satzlänge
  - 32758 Byte bei Dateien mit variabler Satzlänge
  - 32248 Byte bei komprimierter Übertragung (COMPRESS = *BYTE)

Wenn Dateien aus Unix-, Windows- oder POSIX-Systemen in eine DVS-Datei übertragen werden, darf die maximale Satzlänge ohne Angabe des RECORD-SIZE-Parameters im Kommando (2048*b-n) Bytes nicht übersteigen. Dabei ist b der Blockungsfaktor der BS2000-Empfangsdatei. Standard auf NK-PVS-Platten ist b=2, sonst ist der Standard b=1. Auf K-PVS (K für Key) ist n=8, auf NK-PVS (NK für Non Key) ist n=20.

*Lokales z/OS-System*

openFT (z/OS) beachtet bei der Dateiübertragung in der Regel die Satzstruktur (Ausnahme: Übertragung einer unstrukturierten Folge von Binärdaten beim File Transfer mit Unix- oder Windows-Systemen). Bei satzweiser Übertragung darf die maximale Länge von zu übertragenden Sätzen folgende Werte nicht übersteigen:

– 32760 byte bei Dateien mit fixer Satzlänge
– 32752 byte bei Dateien mit variabler Satzlänge (Satzlänge ohne Satzlängenfeld)
– 32248 byte bei komprimierter Übertragung (COMPRESS = *BYTE)

### Übertragung mit transparentem Dateiformat

Eine Besonderheit stellt das transparente Dateiformat dar. Das transparente Dateiformat gibt Ihnen die Möglichkeit, beliebige BS2000-Dateien über verschiedene FT-Plattformen hinweg unter Beibehaltung ihrer ursprünglichen Dateiattribute zu einem BS2000-System durchzureichen. Dieses Verfahren bietet sich beispielsweise an, um BS2000-Dateien von einem Unix-basierten Server oder einem Windows-Server auf BS2000-Systeme zu verteilen. Aus Sicht des Zwischenrechners handelt es sich bei den empfangenen, auf diesem Rechner nicht verwendbaren Dateien um Binärdateien. Auf dem Empfangsrechner werden diese Dateien von openFT (BS2000) dann wieder mit ihren ursprünglichen Attributen eingerichtet.

*Besonderheiten in z/OS*

openFT (z/OS) kann als Zwischenspeicher für BS2000-Dateien im transparenten Dateiformat dienen.

openFT bietet keine unmittelbare Möglichkeit, z/OS-Dateien formattreu ("transparent") über von z/OS verschiedene FT-Plattformen zu übertragen. Sie können jedoch Dateien mit dem TSO-Kommando XMIT in ein neutrales Format packen, und sie in diesem Format als Binärdateien mit einer fixen Satzlänge von 80 byte übertragen. Das geht z.B. durch die Angabe von -r=f80 im ft-Kommando eines openFT im Windows oder auf einem Unix-System. Im Zielsystem kann die Datei mit dem TSO-Kommando RECEIVE wieder entpackt werden.

### Heterogene Übertragung von PAM- und ISAM-Dateien

Sie können BS2000-PAM-Dateien auf ein Fremdsystem wie z.B. ein Unix- oder Windows-System oder ein z/OS auslagern und wieder ins BS2000 zurück holen und dort als PAM-Datei ablegen. Die Initiative für diesen Auftrag kann auch im Fremdsystem sein. Außerdem können Sie ISAM-Dateien aus einem BS2000-System auf ein Fremdsystem übertragen. Voraussetzung ist in allen Fällen, dass auf dem Fremdsystem openFT ab V11 läuft.

Dazu gehen Sie wie folgt vor:

- PAM-Datei vom BS2000 ins Fremdsystem übertragen
  Geben Sie beim Übertragungsauftrag als Zielformat „sequentiell“ an. Der Dateiinhalt wird ohne Dateiende-Marke des C-Laufzeitsystems übertragen, Leerblöcke werden initialisiert.
  Der Inhalt des PAMKEY und die Architektur des Pubsets werden nicht an das Zielsystem übertragen.
- Binärdatei aus einem Fremdsystem im BS2000 als PAM-Datei ablegen
  Geben Sie beim Übertragungsauftrag als Dateiformat „binär“ und als Zielformat „Block-strukturiert“ an. Alle Blöcke bis auf den letzten werden vollständig mit Daten gefüllt, an das Ende wird die Dateiende-Marke des C-Laufzeitsystems angehängt.

  – Wenn Sie PAM-Dateien auslagern und wieder ins BS2000 zurückholen, dann sollte das ursprüngliche Pubset und das Ziel-Pubset möglichst die gleichen Block-Eigenschaften besitzen. Ist dies nicht der Fall oder ist der Inhalt der PAM-Keys für die Datei wesentlich, so muss damit gerechnet werden, dass die Zieldatei unbrauchbar wird.
  Bei PLAM-Bibliotheken ist eine formaterhaltende Übertragung im Allgemeinen möglich. Wenn das Ziel-Pubset NK4 ist, dann muss die PLAM-Bibliothek mit PAMCONV konvertiert werden.

  – Wird eine ausgelagerte PAM-Datei in ein BS2000-System mit openFT V10 zurückgeholt, dann wird sie als sequentielle Datei mit undefiniertem Satzformat abgelegt, bei openFT mit Version < V10 wird der Auftrag abgelehnt.

  – Es ist nicht möglich eine PAM-Datei durch das Holen einer Datei aus dem Fremdsystem zu erweitern.
- ISAM-Datei ins Fremdsystem übertragen
  Geben Sie beim Übertragungsauftrag als Zielformat „sequentiell“ an. Die ISAM-Schlüssel sind Bestandteil der gelesenen Sätze und werden daher mit übertragen. Sie verlieren jedoch ihre Funktion als Index-Schlüssel. Das Satzformat der Zieldatei entspricht dem Satzformat der ISAM-Datei. Das verwendete Format ist kompatibel mit FTP-BS2000.

### 4.3.6 Übertragung von Dateiverzeichnissen (Unix- und Windows-Systeme)

Auf Unix- und Windows-Systemen können Sie mit einem Auftrag auch ein komplettes Dateiverzeichnis zwischen Partnersysteme übertragen. Dabei haben Sie folgende Möglichkeiten:

- Synchrones und asynchrones Senden
- Synchrones und asynchrones Holen
- Modus "Neu Erzeugen" und "Überschreiben"

Einschränkungen: Die Verzeichnisübertragung wird nur für das openFT-Ptotokoll unterstützt. Das Erweitern vorhandener Dateien sowie Vorverarbeitung und Nachverarbeitung sind nicht möglich.

### 4.3.7 Ausgelagerte Dateien

openFT kann auf ausgelagerte (migrierte) Dateien in BS2000- und z/OS-Systemen zugreifen. Damit kann man sich die Eigenschaften solcher Dateien ansehen, die Dateien übertragen, löschen oder überschreiben. Voraussetzung ist, dass im betroffenen System openFT ab V10 eingesetzt wird. Für die Mainframe-Systeme gelten dabei folgende Besonderheiten:

- Bei BS2000-Systemen muss die Datei eine DMS-Datei sein. Einzelne Elemente einer migrierten Bibliothek können nicht direkt übertragen werden. Hierzu muss zuerst die migrierte Bibliothek eingelesen werden. Dieses kann z.B. über Vor- bzw. Nachverarbeitung oder /EXEC-REM-CMD bzw. *ftexec* erfolgen.
- Bei z/OS-Systemen muss mindestens z/OS ab V1.7 eingesetzt werden, da erst mit dieser Version die notwendigen Werte an der Systemschnittstelle zurückgeliefert werden.

## 4.4 Übertragung von 7 Bit-, 8 Bit- und Unicode-Dateien

In Rechnern mit unterschiedlichen Betriebssystemen werden die einzelnen Zeichen, Buchstaben und Ziffern auf verschiedene Arten intern dargestellt ("codiert"). Außerdem können auf den verschiedenen Rechnern unterschiedliche Zeichensätze eingesetzt werden.

Unterschiedliche Zeichensätze sind bei der Übertragung von Textdateien von Bedeutung, nicht jedoch bei Übertragung in anderen Dateiformaten (binär, transparent, ...), da openFT den Dateiinhalt dann nicht konvertiert.

Die Umcodierung von Textdateien wird durch openFT vorgenommen.

Je nach verwendetem Zeichensatz wird der Inhalt einer Textdatei unterschiedlich interpretiert und am Bildschirm oder auf dem Drucker entsprechend dargestellt.

openFT bietet die Möglichkeit, Textdateien sowohl verschiedene Einbyte-Zeichensätze (7 Bit und 8 Bit) als auch Mehrbyte-Zeichensätze (Unicode) zuzuordnen.

**Hinweis für z/OS:** Ein vollständiges PO/PDSE -Dataset wird im Sendesystem von openFT nicht umcodiert, da ansonsten die Steuerinformationen in der Zieldatei eventuell nicht mehr korrekt sind.

### 4.4.1 Code-Tabellen und Coded Character Sets (CCS)

Für openFT-Partner ab V10 wird das Konzept der sogenannten „Coded Character Sets“ (CCS) aktiv unterstützt. Ein CCS definiert einen Zeichensatz und die Codierung dieser Zeichen in der Datei. Einem CCS ist ein bis zu 8 Zeichen langer Name zugeordnet, über den der CCS angesprochen werden kann.

Der Benutzer kann bei der Übertragung von Textdateien jeweils einen CCS für die Codierung der Datei im lokalen System und fernen System angeben.

Häufig verwendete Coded Character Sets sind

| | |
|---|---|
| ISO88591 | Zeichenvorrat gemäß Definition der ISO-Norm 8859-1, ASCII orientierte Codierung gemäß ISO-Norm 8859-1 |
| EDF041 | Zeichenvorrat gemäß Definition der ISO-Norm 8859-1, EBCDIC orientierte Codierung gemäß Fujitsu-Definition DF04-1 |
| IBM1047 | Zeichenvorrat gemäß Definition der ISO-Norm 8859-1. IBM1047 ist eine EBCDIC orientierte Codierung gemäß IBM-Definition IBM1047 und wird in z/OS-Systemen als Standard verwendet. |
| UTF8 | Zeichenvorrat ist Unicode, Codierung ist die im Unicode Standard festgelegte Mehrbyte Codierung UTF-8 |

UTF16 Zeichenvorrat ist Unicode, Codierung ist die im Unicode Standard festgelegte 16-Bit Codierung UTF-16

CP1252 Zeichenvorrat ist eine von Microsoft definierte Obermenge des Zeichenvorrats der ISO-Norm 8859-1, die ASCII orientierte Codierung ist bei den mit ISO8859-1 gemeinsamen Zeichen identisch mit der ISO8859-1 Codierung. Die zusätzlich von Microsoft definierten Zeichen (u.a. das Euro-Symbol) liegen im von ISO8859-1 nicht benutzten Codebereich 0x80-0x9F.

**CCS bereitstellen**

- Im BS2000-System werden die CCS über XHCS definiert und zur Verfügung gestellt. Der Standard-CCS für das System (HOSTCODE) wird vom BS2000-Systemverwalter festgelegt. Zusätzlich kann dieser einer Benutzerkennung einen von HOSTCODE abweichenden Standard-Anwender-Zeichensatz zuordnen.
- Auf den anderen openFT-Plattformen ab V10 werden die gängigen CCS mit openFT ausgeliefert, der Standard-Zeichensatz wird vom FT-Verwalter per Betriebsparameter festgelegt.

### 4.4.2 CCS beim Übertragungsauftrag angeben

Beim Übertragen von Textdateien können Sie sowohl für das lokale System als auch für das ferne System einen auftragsspezifischen CCS angeben, mit dem die lokale und/oder ferne Datei gelesen oder geschrieben wird.

Der ferne CCS-Name wird nur für das openFT-Protokoll und für Partner ab V10 unterstützt.

Ist die lokale oder die ferne Datei eine BS2000-Datei, der über den Katalogeintrag schon ein CCS-Name zugewiesen ist, dann dürfen Sie keinen davon abweichenden CCS-Namen angeben.

Wird der lokale bzw. der ferne CCS-Name weggelassen, dann gelten die Standardeinstellungen des jeweiligen Systems:

– openFT-Betriebsparameter im Unix-, Windows- oder z/OS-System,

– im BS2000 der CCS des Katalogeintrags der Datei (falls vorhanden), sonst der Systemparameter HOSTCODE.

Im z/OS kann Dateien per Einstellung in der FT-Parameterbibliothek zusätzlich ein bestimmter CCS zugeordnet werden. Details siehe Handbuch "openFT (z/OS) - Installation und Betrieb".

**Achtung!**
Wenn Sie für das Abspeichern der Datei einen Zeichensatz verwenden, der keine Obermenge des in der Datei ursprünglich verwendeten Zeichensatzes ist, geht

Information verloren! Alle Zeichen, die sich in dem neu zugewiesenen Zeichensatz nicht abbilden lassen, werden auf ein Ersatzzeichen abgebildet. Eine solche Konvertierung kann nicht verlustfrei rückgängig gemacht werden!

### 4.4.3 Datenkonvertierung

Die Art der Datenkonvertierung hängt davon ab, welche openFT-Version im Partnersystem eingesetzt wird.

**Datenkonvertierung bei Partnern mit openFT ab V10**

Abhängig von der Codeklasse (ISO 8859 oder DF04) und der Codevariante n (n=1...10, 13, 15) des lokalen CCS sendet openFT ab V10 die Daten in den Codierungen ISO 8859-n, DF04-n bzw. UTF-8.

Dies bewirkt je nach Partnersystem und Codierung Folgendes:

– Dateien in Unix- und Windows-Systemen, denen ein ISO8859n CCS zugeordnet ist, werden bei Sendeaufträgen an Unix- oder Windows-Systeme nicht mehr umcodiert. Bei Übertragungen zwischen Unix- oder Windows-Systemen findet für die Übertragung selbst also gar keine Umcodierung mehr statt, wenn auch für die Zieldateien derselbe ISO8859n CCS zugeordnet wurde.

– Bei Übertragungen von Dateien der Codeklassen ISO 8859 oder DF04 zwischen Unix- und Windows-Systemen und BS2000 oder z/OS wird auf der jeweiligen Empfangsseite umcodiert (falls notwendig).

– UTF-8 Dateien werden auf Empfangsseite umcodiert (falls notwendig). Dateien, denen ein CCS zugeordnet ist, der weder zur Codeklasse ISO 8859 noch zu DF04 gehört, werden auf Senderseite nach UTF-8 und auf Empfangsseite gegebenenfalls auf den CCS für die Zieldatei umcodiert.

– UTF-16 Dateien werden auf Sendeseite in UTF-8 umcodiert und auf Empfangsseite (falls gefordert) in UTF-16.

– Von openFT erzeugte UTF-16 Dateien haben das Endian-Modell und die Zeilentrennkonvention (LF bzw. CRLF) der jeweiligen Plattform.

– Von openFT erzeugte UTF-8 Dateien haben die Zeilentrennkonvention (z.B. LF oder CRLF) der jeweiligen Plattform.

### Datenkonvertierung bei Partnern mit openFT < V10

Die übertragenen Daten sind in DF04-n codiert. D.h. bei Dateiübertragungen mit openFT-Partnern werden die Daten im EBCDIC übertragen (entspricht CCS DF04-n). EBCDIC wird beispielsweise in BS2000-Systemen verwendet. Bei Übertragungen mit openFT-Partnern auf Unix- oder Windows-Systemen werden Textdateien im Partnersystem umcodiert.

### Hinweis für Unix- und Windows-Systeme

openFT codiert Textdateien bei Übertragungen mit openFT-Partnern wie folgt um:

– beim Holen einer Datei in ein Unix- oder Windows-System von EBCDIC nach ISO 8859,

– beim Senden einer Datei aus einem Unix- oder Windows-System von ISO 8859 nach EBCDIC.

Über ISO 8859 hinausgehende Sonderzeichen oder Ersatzdarstellungen werden bei der Umcodierung nicht umgesetzt. Dateien, die solche Zeichen enthalten, sollten Sie als Binärdateien übertragen und von einer selbst zu erstellenden Umcodierungsroutine umsetzen lassen.

Bei Datenübertragungen, die über die FTAM-Funktionalität abgewickelt werden, wird bei Kopplungen mit Fremdprodukten und openFT-Partnern < V10 davon ausgegangen, dass für die Übertragung und für die lokale Datei ISO 8859 verwendet wird. Es wird daher lokal nicht umcodiert.

*Textformat*

Beim Senden geht openFT davon aus, dass die zu versendende Datei eine reine ISO 8859-Textdatei ist, die durch Zeilenende-Kennzeichen in Sätze strukturiert ist.

In bestimmten Situationen wird eine Konvertierung vorgenommen, d.h. Tabulatorzeichen werden in Leerzeichen expandiert und Zeilenende-Kennzeichen werden eliminiert. Abhängig von der Situation (Inbound, Outbound) und den beteiligten Partnern gilt Folgendes:

– Inbound-Aufträge:

  Auf der Inbound-Seite wird grundsätzlich nicht im Unix- oder Windows-System konvertiert, weder beim Senden noch beim Empfangen.

– Outbound-Aufträge, die in einem Unix- oder Windows-System gestellt werden:

  Beim Empfangen wird grundsätzlich nicht konvertiert.

  Beim Senden kann auftragsspezifisch konvertiert werden *(ft -tb=* und *ncopy -tb=*, TabExpansion). Standardmäßig wird beim Senden zu einem BS2000-, OS/390- oder z/OS-Partner konvertiert, sonst nicht.

– Outbound-Aufträge, die in einem BS2000s-, OS/390- oder z/OS-System gestellt werden:

  Beim Senden wird grundsätzlich nicht konvertiert.

  Beim Empfangen wird partnerabhängig konvertiert, d.h. bei einem Unix- oder Windows-Partner wird konvertiert und bei einem BS2000-, OS/390- oder z/OS-Partner wird nicht konvertiert.

*Binärformat*

openFT geht davon aus, dass die zu übertragende Datei eine unstrukturierte Folge von Binärdaten enthält. Im Empfangssystem wird eine Datei mit undefinierter Satzlänge erzeugt. Die Binärdaten bleiben erhalten.

*Benutzerformat*

Beim Senden geht openFT davon aus, dass die zu versendende Datei durch Längenfelder in Sätze strukturiert ist. Die ersten beiden Bytes eines jeden Satzes müssen dessen Länge inklusive der Länge des Satzlängenfeldes enthalten. Beim Holen erzeugt openFT diese Längenangaben entsprechend der im fernen System bestehenden Satzlängen. Der Inhalt der Sätze wird als Binärdaten behandelt, d.h. es wird nicht umcodiert.

Die Satzstruktur und die Binärdaten bleiben bei der Übertragung erhalten. Die Satzlängenfelder werden in einem Windows-System mit dem höchstwertigen Byte zuerst abgespeichert.

Für FTP-Partner ist es nicht sinnvoll, das Benutzerformat zu verwenden, da die Satzstruktur verloren geht. Zwischen FTAM-Partnern wird ein anderer Mechanismus verwendet (siehe Abschnitt „Virtueller Dateispeicher“ auf Seite 107).

### 4.4.4 XHCS-Unterstützung durch openFT auf BS2000-Systemen

Mit XHCS können innerhalb eines BS2000-Systems verschiedene Zeichensätze zur selben Zeit verwendet werden. openFT kann mit Hilfe von XHCS-Informationen die aktuelle Codierung von Daten erkennen. Abhängig von Art und Umfang der Informationen setzt openFT vor bzw. nach der Übertragung einer Datei als Textdatei die Daten mit Hilfe der Umsetztabellen von XHCS in einen Code um, der im Zielsystem verarbeitet werden kann.

Details siehe Handbuch "openFT (BS2000) - Installation und Betrieb".

## 4.5 Zeichencode-Unterstützung für Dateinamen und Kommandos auf Unix- und Windows-Systemen

Auf Unix- und Windows-Systemen unterstützt openFT unterschiedliche Zeichencodierungen bei der Angabe von

- fernen Dateinamen und Verzeichnisnamen,
- Vor-, Nach- und Folgeverarbeitungen,
- fernen Kommandos.

Für diese Parameter wird zwischen dem **transparanten Modus** und dem **Zeichenmodus** unterschieden.

### Transparenter Modus

Dateinamen etc. werden in einer festen binären Verschlüsselung unabhängig von lokalen Zeichencode-Einstellungen gesehen. Eine Codeumsetzung erfolgt lediglich zwischen EBCDIC DF.04-1 (BS2000), IBM1047 (z/OS) und ISO8859-1 (Unix, Windows). Verwenden die beteiligten Systeme zur Darstellung ihrer Dateinamen unterschiedliche Codevarianten, so können sich die Zeichen ändern.

Der transparente Modus ist kompatibel zu den Vorversionen von openFT.

*Hinweis für Window-Systeme*

In Windows-Systemen besteht für ferne Dateinamen und Kommandostrings in fernen Folgeverarbeitungen im transparenten Modus eine gewisse Abhängigkeit vom lokal eingestellten ANSI-Code, da in früheren openFT-Versionen an der Windows-Systemanbindung Dateinamen und Kommandos im lokalen ANSI-Code gesehen wurden. Unabhängig vom gewählten Modus existiert diese Abhängigkeit auch für die meisten übrigen Parameter wie zum Beispiel die ferne Zugangsberechtigung.

### Zeichenmodus

Dateinamen etc. werden in ihrer Zeichendarstellung gesehen. Ein Ä in einem fernen Dateinamen wird beispielsweise im Zeichenmodus auch im Partnersystem als Ä verstanden, auch wenn dort eine andere Systemcodierung eingerichtet bzw. eine andere lokale Codierung eingestellt ist. Ist das Ä im Zeichenvorrat des Partnersystems nicht enthalten, wird das entsprechende Kommando abgebrochen (z.B. "Syntaxfehler im resultierenden Dateinamen"). Der Zeichenmodus setzt voraus, dass der Partner openFT V12.1 oder höher ist.

Das FTP-Protokoll unterstützt den Zeichenmodus nicht.

*Hinweis für Windows-Systeme*

In Windows können im Zeichenmodus ferne Dateinamen und Kommandostrings in fernen Folgeverarbeitungen unabhängig vom lokal eingestellten ANSI-Code angegeben werden; der Zeichenvorrat ist gegenüber dem von Windows-Dateinamen (UCS-2) nicht eingeschränkt. Für lokale Dateinamen gilt dies sogar unabhängig vom gewählten Modus.

### 4.5.1 Eigenschaften des Zeichenmodus

Die Übertragung der Pfadnamen zwischen den Partnersystemen erfolgt im Zeichenmodus in UTF-8, so dass auch Zeichen außerhalb des bisher von openFT unterstützten Zeichenumfangs übertragen werden und somit auch exotische Pfadnamen angesprochen werden können.

#### Anzeige von fernen Dateinamen

Wird ein Kommando zum Holen von Dateiinformationen im Zeichenmodus ausgeführt, so werden auch die geholten Dateinamen aus dem Zeichencode des Partnersystems in den lokalen Zeichencode umgesetzt. Dateien, deren Namen keiner gültigen Codierung nach der Einstellung im Partnersystem entsprechen, werden beim Auflisten von Verzeichnissen übersprungen. So werden zum Beispiel Dateinamen aus einem Unix-System bei eingestelltem UTF-8 (*ftmodo -fnccs=utf8*) nicht angezeigt, die keiner gültigen UTF-8-Codierung entsprechen. Im transparenten Modus werden Dateinamen hingegen nach den bisherigen, von openFT vergebenen Regeln entsprechend den beteiligten Betriebssystemen umgesetzt bzw. als Bytestrings beibehalten. Auf Windows werden Dateinamen hierbei übersprungen, die nicht dem Zeichenvorrat des eingestellten ANSI-Codes entsprechen.

#### Hinweise für Unix-Systeme

Der Zeichencode eines Unix-Partnersystems bestimmt sich auf Inbound-Seite aus der neuen openFT-Option FNCCS, die der openFT-Verwalter dieses Systems per Betriebsparameter einstellt (z.B. über das Kommando *ftmodo -fnccs)*. Hierfür sollte er die vorherrschende lokale Codierung angeben, z.B. UTF8 oder ISO8859-1.

Auf Outbound-Seite wird die Zeichendarstellung für die Angabe ferner Dateinamen, Vorverarbeitungskommandos etc. aus der "LOCALE" bzw. der Umgebungsvariablen LANG abgeleitet (z.B. ISO8859F bei de_DE@euro, ISO88597 bei el_GR, oder UTF8 bei de_DE.utf8). Wenn Kommandos über Terminal oder Konsole eingegeben werden, sollte auch die Terminal-Anzeige dieser Einstellung entsprechen. Die aus der "LOCALE" bzw. der LANG-Variablen abgeleitete Codetabelle muss in openFT integriert sein.

Sowohl bei der Festlegung von FNCCS als auch bei der LANG-Einstellung sollten Zeichencodes vermieden werden, bei denen die Codierung des ISO646-Zeichenvorrats von der ISO8859-1-Codierung abweicht (z.B. ISO646de, EDF041, UTF16). Andernfalls würden Vorverarbeitung, *ftexec*, *ftadm* oder gewöhnliche Dateizugriffe im Zeichenmodus nicht richtig funktionieren.

**Hinweise für Windows-Systeme**

Bei aktuellen Windows-Systemen sind die Pfadnamen immer in UCS-2 (einer Untermenge von UTF-16) im Dateisystem hinterlegt, sodass hier keine zusätzliche openFT-Option nötig ist. In der Auftragsbeschreibung und beim Zeichenmodus auch bei der Übertragung sind die Pfadnamen in UTF-8 codiert.

**Hinweise für B2000- und z/OS-Systeme**

Im BS2000 und z/OS wird auch im Zeichenmodus von festen Zeichencodes ausgegangen (EBCDIC DF.04-1, bzw. IBM1047).

## 4.5.2 Empfehlungen zur Verwendung des Transparent- und Zeichenmodus

Werden für Dateinamen und Folgeverarbeitung stets ISO646-Zeichen verwendet, so kann sowohl mit transparentem als auch mit Zeichenmodus gearbeitet werden. Ansonsten beachten Sie bitte die folgenden Abschnitte.

### 4.5.2.1 Verwendung des transparente Modus

- Beschränkt sich der Zeichenvorrat von Dateinamen auf den von ISO8859-1, und werden auch entsprechende Code-Einstellungen auf den Rechnern verwendet (z.B. LANG=de_DE auf Unix und CP1252 auf Windows), so empfiehlt sich der transparente Modus. Ist der Zeichenvorrat bei einer Anwendung des openFT-Produkts auf eine Codevariante von ISO8859 (z.B. ISO8859-7 mit griechischen Zeichen) beschränkt, so kann ebenfalls mit dem transparenten Modus gearbeitet werden.
- Solange es keine Probleme mit der Zeichendarstellung von Dateinamen gibt (z.B. infolge einer Ausweitung auf zusätzliche Sprachräume), empfiehlt es sich, in bestehenden Anwendungen von openFT den transparenten Modus beizubehalten.

Sind in einer Anwendung von openFT-Instanzen aus mehreren Sprachräumen miteinander gekoppelt, die jeweils nationale Code-Einstellungen verwenden (z.B. ISO8859-1 und ISO8859-7), so kann die Zeichendarstellung von Dateinamen variieren, wenn der transparente Modus benutzt wird.

#### 4.5.2.2 Verwendung des Zeichenmodus

- Steht eine Ausweitung des Dateinamens-Zeichenvorrats von ISO646 auf einen 8-Bit-Zeichencode an, so sollte vorausschauend ein Umstieg auf den Zeichenmodus in Erwägung gezogen werden, da dieser von ISO646 ausgehend wesentlich leichter möglich ist als dann, wenn in der Zwischenzeit auf einen 8-Bit-Zeichencode umgestiegen wurde.
- Wird in den Instanzen einer openFT-Anwendung UTF-8 als Zeichencode für Dateinamen verwendet, so empfiehlt es sich, den Zeichenmodus zu wählen. Der Zeichenmodus kann am effektivsten genutzt werden, wenn alle Instanzen ihre Dateinamen in UTF-8 codieren (auf Windows-Instanzen gilt dies prinzipiell schon durch Verwendung des Zeichenmodus).

**ACHTUNG!**
Sind in einer Anwendung von openFT Instanzen aus mehreren Sprachräumen miteinander gekoppelt, und werden in bestimmten Instanzen anstelle von UTF-8 "nationale" oder "regionale" Code-Einstellungen wie ISO8859-1 für Dateinamen verwendet, so werden Dateiübertragungsaufträge im Zeichenmodus abgelehnt, deren Angaben für den fernen Dateinamen im dort eingestellten Code nicht abbildbar sind.

**Hinweise für Windows-Systeme**

- Der Wechsel zum Zeichenmodus ist für Dateisysteme auf Windows unproblematisch.
- Auf Windows werden Dateinamen in den Ausgaben von Kommandos wie *ftshwl*, *ftshwr*, *ftshwp* etc. in Unicode dargestellt. *ftshwl | more* ersetzt jedoch Zeichen außerhalb der lokal eingestellten Codetabelle durch ? oder andere Ersatzzeichen.
  Für größere Ausgaben empfiehlt es sich, OPENFTOUT=UTF8 als Umgebungsvariable zu setzen, die Ausgabe auf eine Datei umzuleiten und diese dann anzuschauen mit *ftedit -ro -ccs=utf8*.
- Wenn openFT auf Windows Dateien auf einem Verzeichnis anlegt, das nicht auf einem lokalen Laufwerk liegt (z.B. ein per SAMBA angebundenes Verzeichnis), so kann dies zu Zeichenersetzungen beim Dateinamen führen, die auch beim lokalen Kopieren einer Windows-Datei auf dieses Verzeichnis entstehen würden.

**Hinweise für Unix-Systeme**

- Vorsicht ist geboten bei einem Wechsel der Zeichencodierung von Dateinamen auf einem Unix-System, zum Beispiel anlässlich eines Umstiegs von transparentem Modus auf Zeichenmodus. Dies kann zu uneinheitlich codierten Dateibäumen führen (wie UTF-8 codierte Dateinamen gemischt mit ANSI-Dateinamen in einem alten Verzeichnis, das im Namen noch einen ANSI-Umlaut enthält), mit denen nicht nur openFT Schwierigkeiten haben dürfte.

Es müssten alle Datei- und Verzeichnisnamen, Referenzen auf Dateinamen etc., auf die nach dem Wechsel noch zugegriffen werden soll, auf die neue Codierung umbenannt werden (z.B. ISO8859-1 -> UTF-8). Nur wenn bisher ausschließlich ISO646-Zeichen in Dateinamen verwendet wurden, ist ein Umstieg ohne diesen Zusatzaufwand möglich.

- Sofern openFT auf Unix für eine Anwendung nur auf Dateien inbound zugreift, die in einem definierten Verzeichnis liegen, kann dieses in einer FTAC-Zugangsberechtigung als Dateinamenspräfix festgelegt werden. Wenn openFT bei Aufträgen im Zeichenmodus diese Zugangsberechtigung verwendet, genügt es, wenn die Dateinamen relativ zu diesem Präfix der Code-Einstellung entsprechen; für den Verzeichnisnamen gibt es in diesem Fall keine Beschränkungen. Er dürfte zum Beispiel auch in ISO8859-1 codierte Umlaute enthalten, selbst wenn *ftmodo -fnccs=utf8* eingestellt ist.

### Zugangsberechtigungen

Vorsicht ist geboten bei der Definition von Zugangsberechtigungen mit Zeichen außerhalb des ISO646-Zeichenvorrats. Bei Dateiübertragungsaufträgen und File-Management-Aufträgen wird die angegebene Zugangsberechtigung grundsätzlich entsprechend dem transsparenten Modus verarbeitet, auch wenn (für Dateinamen) Zeichenmodus gewählt wurde. Bei der Festlegung einer Zugangsberechtigung auf einem lokalen Unix-System wie auch über Fernadministration im Zeichenmodus wird diese jedoch im dort eingestellten lokalen Zeichencode angelegt, bei Fernadministration im Zeichencode, der mit *ftmodo -fnccs=…* definiert ist.

### Trace-Auswertungen

Trace-Auswertungen können bei Dateinamensangaben auch UTF-8-Strings enthalten. Es kann deshalb zweckmäßig sein, einen ausgewerteten Trace mit *ftedit -ro -ccs=utf8* anzuschauen, um diese Dateinamen in der korrekten Zeichendarstellung zu sehen. Bytefolgen in der Trace-Auswertung, die nicht korrektem UTF-8 entsprechen, werden allerdings nicht unterdrückt oder durch ?, _ oder ähnlichem ersetzt, sondern über eine Heuristik als ISO8859-1-Zeichen ausgewertet. Dadurch geht die Information verloren, ob ein Umlaut in ISO oder in UTF-8 codiert ist. Um dies festzustellen, ist es sinnvoll, auch mal ohne den Schalter *-ccs=utf8* den Trace anzuschauen.

## 4.6 Angaben zum fernen System

Mit den Angaben zum fernen System definieren Sie das Partnersystem und geben Ihre Zugangsberechtigung zu einer Benutzerkennung im Partnersystem bekannt.

Details zum Partnerkonzept und zu Partnersystemen finden Sie im Kapitel „Partnerkonzept“ auf Seite 51.

### 4.6.1 Partnersystem

Das Partnersystem ist das ferne System, mit dem Dateien ausgetauscht werden sollen. Sie sprechen das Partnersystem über einen Partnernamen oder seine Partneradresse ("**dynamische Partner**") an.

Der FT- Verwalter kann die Verwendung von dynamischen Partnern aus Sicherheitsgründen abschalten. In diesem Fall können Sie nur Partnernamen aus der Partnerliste verwenden.

#### Partnername

Ein Partnername ist ein maximal 8 Zeichen langer Name, den der FT-Verwalter vergibt, wenn er ein Partnersystem in die Partnerliste aufnimmt. Das sollte vor allem mit Partnersystemen geschehen, mit denen häufig kommuniziert wird.

#### Partneradresse

Wenn der FT-Verwalter keinen Partnernamen vergeben hat oder Ihnen dieser Name nicht bekannt ist, können Sie einen Partnerrechner über die Partneradresse ansprechen. Eine Partneradresse hat folgenden Aufbau:

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

*host* (= Rechnername des Partnersystems) muss immer angegeben werden, alle anderen Angaben sind optional, siehe auch Abschnitt „Partneradressen“ auf Seite 52. In vielen Fällen werden sie durch Standardwerte abgedeckt, so dass der Rechnername als Partneradresse ausreicht.

Die Adressbestandteile werden in den jeweiligen openFT-Handbüchern "Kommandoschnittstelle" erklärt.

### 4.6.2 Zugangsberechtigung

Die Zugangsberechtigung kann angegeben werden als:

- Login-/LOGON-Berechtigung bestehend aus Benutzerkennung, Abrechnungsnummer und Passwort. Diese Werte sind systemabhängig.
- FTAC-Zugangsberechtigung, die betriebssystemunabhängig festgelegt werden kann und zusätzlich einen höheren Zugangsschutz bietet. Die FTAC-Zugangsberechtigung muss im Partnersystem definiert werden.

Der Inbound-Zugriff über Standard-FTP-Client ist ebenfalls möglich.

Weitere Details werden in den jeweiligen openFT-Handbüchern "Kommandoschnittstelle" erklärt.

## 4.7 Optionen bei der Dateiübertragung

openFT bietet Ihnen die Möglichkeit, optional weitere Einstellungen für die Dateiübertragung festzulegen. Es können individuelle Satzlängen definiert, Schreibregeln und Dateikomprimierung vereinbart und Konditionen für die Ergebnismitteilung sowie Zugriffsmodalitäten für FTAM-Partner festgelegt werden.

### 4.7.1 Maximale Satzlänge

Unter der maximalen Satzlänge versteht man die Länge des längsten Satzes (Nettosatzlänge in Bytes) ohne Berücksichtigung von Satzlängenfeldern.

Da openFT auch Dateien in Unicode-Zeichensätzen übertragen kann, muss man zwischen Länge in Bytes und Länge in Zeichen unterscheiden.

Bei BS2000- und z/OS-Dateien ist die maximale Satzlänge als Dateiattribut im Katalog hinterlegt (bei variabel langen Sätzen mit einem Aufschlag von 4 für das Satzlängenfeld).

In Unix- und Windows-Systemen sowie in POSIX-Systemen können Sie die maximale Satzlänge Ihrer Datei, die Sie als Text- oder satzstrukturierte Binärdatei (Benutzerformat) übertragen wollen, individuell einstellen. Die vorgegebene maximale Satzlänge muss mindestens so groß wie die größte tatsächlich vorhandene sein, da sonst der FT-Auftrag nicht ausgeführt werden kann.

Beachten Sie dabei, dass die maximale Satzlänge nicht nur auf Sende- und auf Empfangsseite, sondern auch während der Übertragung eingehalten werden muss. Für die Übertragung im UTF8-Zeichensatz (siehe Abschnitt „Datenkonvertierung“ auf Seite 83) kann während der Übertragung eine größere Satzlänge benötigt werden als auf Sende- und Empfangsseite.

### 4.7.2 Schreibregel

Mit der Option "Schreibregel" definieren Sie, wie bei der Dateiübertragung mit der Zieldatei verfahren werden soll. Diese Option kann auch über FTAC definiert werden. Es bestehen folgende Möglichkeiten:

– Dateien zu überschreiben, d.h. vorhandene Dateien werden, sofern die Dateimerkmale dies zulassen, überschrieben bzw. nicht vorhandene neu eingerichtet,

– Dateien zu erweitern, d.h. vorhandene Dateien werden, sofern die Dateimerkmale dies zulassen, am Dateiende erweitert bzw. nicht vorhandene neu eingerichtet,

– Dateien nicht zu überschreiben, in diesem Fall werden vorhandene Dateien keinesfalls überschrieben, sondern der FT-Auftrag mit entsprechender Meldung abgebrochen. Ist die angegebene Zieldatei nicht vorhanden, wird sie neu eingerichtet.

In z/OS ist die genaue Auswirkung der Option "Schreibregel" (Operand WRITE-Mode) auch abhängig von der Art der Empfangsdatei (PS-Dataset, Member eines PO-Datasets usw.). Dies ist im Einzelnen beschrieben im Handbuch "openFT (z/OS) - Kommandoschnittstelle".

### 4.7.3 Komprimierte Dateiübertragung

Dateien können komprimiert übertragen werden, um die Übertragungszeiten zu verkürzen und so eventuell Übertragungskosten zu sparen. Dabei ist zu beachten, dass die Komprimierung im sendenden Rechner und die Dekomprimierung im empfangenden Rechner CPU-Last erzeugt.

openFT kennt zwei Komprimierungsmethoden - die Zip-Komprimierung (mit openFT-Partnern ab V10) und die Byte-Komprimierung. Mit beiden lässt sich die zu übertragende Datenmenge reduzieren. Durch das Komprimieren und Dekomprimieren erhöht sich der CPU-Bedarf und damit die Zeit, die ein Auftrag vor und nach der eigentlichen Datenübermittlung benötigt.

Auf "schnellen" Leitungen (ab etwa 10 Mbit) wird die Gesamtlaufzeit eines Auftrages durch eine Komprimierung normalerweise nicht verbessert. Auf "langsamen" Leitungen (unterhalb 1 Mbit) kann die Zip-Komprimierung Performancevorteile bringen. Byte-Komprimierung lohnt sich, wenn Dateien übertragen werden sollen, in denen zahlreiche Byte-Wiederholungen vorkommen (z.B. Listen mit Blanks zur Spaltenausrichtung, Dumps mit zahlreichen Nullen). Wenn das Partnersystem keine Komprimierung unterstützt, überträgt openFT die Datei unkomprimiert. openFT-FTP unterstützt die Byte-Komprimierung, wie in RFC959 beschrieben.

Die komprimierte Dateiübertragung zu FTAM-Partnern wird nicht unterstützt.

### 4.7.4 Verschlüsselte Dateiübertragung

openFT kann auf Anforderung des Nutzers Daten verschlüsselt versenden (siehe auch Abschnitt „Verschlüsselung der Dateiinhalte“ auf Seite 131).

Die Option Verschlüsselung ist aus rechtlichen Gründen nicht in allen Ländern verfügbar, d.h. die verschlüsselte Dateiübertragung mit ausländischen Partnern ist nicht in jedem Fall gewährleistet.

Von openFT verschlüsselte Daten können über das FTP-Protokoll nur Outbound und nur mit Secure-FTP-Partnern ausgetauscht werden. Mit FTAM-Partnern können keine per openFT verschlüsselten Daten ausgetauscht werden.

Verschlüsselte Dateiübertragung setzt voraus, dass auf openFT-Seite immer openFT-CR installiert sein muss, d.h. auch im Partnersystem, wenn dort ein openFT läuft.

### 4.7.5 Übertragung von Schutzattributen zwischen BS2000-Systemen

Wenn der Partner ein BS2000-System mit openFT ab V11.0 ist, dann können DMS-Dateien so übertragen werden, dass zusätzlich zu den Standard-Dateiattributen auch die Schutzattribute USER-ACCESS, ACCESS, BASIC-ACL, EXPIRATION-DATE, FREE-FOR-DELETION und DESTROY übermittelt werden.

Die Übertragung der Schutzattribute ist nur über das openFT-Protokoll und nicht im transparenten Modus möglich. Die Zieldatei muss neu angelegt oder überschrieben werden und darf keine Dateigeneration sein.

### 4.7.6 Ergebnismitteilung

Der Auftraggeber eines File Transfer-Auftrags kann sich über das Ergebnis seines Auftrags informieren lassen. Hierzu eignet sich insbesondere die Logging-Funktion, die auf allen Plattformen einheitlich verfügbar ist.

Weitere Möglichkeiten der Ergebnismitteilung sind von der Plattform abhängig:

– Auf BS2000-Systemen sendet das FT-System asynchrone Meldungen an den Benutzer, die das Ergebnis der Dateiübertragung mitteilen. Voraussetzung ist, dass der Benutzerprozess, der den FT-Auftrag gegeben hat, noch aktiv ist und asynchrone Meldungen zugelassen sind.

– Auf z/OS- und BS2000-Systemen wird auf Wunsch des Auftraggebers eine Datei angelegt, die abhängig von Erfolg oder Misserfolg automatisch ausgedruckt werden kann.

– Auf Unix-Systemen kann die Ergebnismitteilung abhängig vom Ergebnis im Postkorb des Auftraggebers abgelegt werden.

Details zu z/OS finden Sie im Handbuch "openFT (z/OS) - Kommandoschnittstelle".

### 4.7.7 Zugriffsmodus für FTAM-Partner

Für den File Transfer mit FTAM-Partnern besteht die Möglichkeit, FTAM-spezifische Dateiattribute zu definieren. Die FTAM-Dateiattribute, die den Dateityp beschreiben, müssen identisch mit den Angaben zum Dateityp im Dateiübertragungsauftrag sein. Die entsprechenden Attribute werden in Abschnitt „Besonderheiten beim File Transfer mit FTAM-Partnern“ auf Seite 107ff vorgestellt.

### 4.7.8 Vorverarbeitung und Nachverarbeitung

Die Funktionen "Vorverarbeitung" und "Nachverarbeitung" erlauben es, mit Hilfe eines Dateiübertragungs-Auftrags im lokalen und im fernen System beliebige Kommandos (Betriebssystemkommandos, Prozeduren, ...) zu starten. Die Kommandos werden statt des Dateinamens an das entsprechende System übergeben. Dabei gilt Folgendes:

- Der Dateiname muss in Anführungszeichen (") eingeschlossen (Unix- und Windows-Systeme) bzw. als C-String angegeben werden (BS2000-Systeme, z/OS).
- Das erste Zeichen ist ein Pipe-Zeichen ’|‘. Darauf folgen die Kommandos getrennt durch ’;‘ (bzw. ’&’ oder ’&&’ in Windows-Systemen, wobei der Kommandostring in diesem Fall mit *cmd /c* beginnen muss). Die maximale Länge der Vor- bzw. Nachverarbeitungskommandos ist begrenzt durch die maximale Länge des Dateinamens.

  Wenn anstelle des Pipe-Zeichens die Zeichenfolge ’|&’ angegeben wird, dann ist der Übertragungsauftrag wiederanlauffähig, siehe Seite 97.

Die Vorverarbeitung übergibt das Ergebnis an die Standardausgabe des Systems (SYSLST auf BS2000, SYSPRINT auf z/OS, stdout auf Unix-Systemen und Windows), die Nachverarbeitung liest die Daten von der Standardeingabe des betreffenden Systems (SYSIN auf BS2000, SYSTSIN auf z/OS, stdin auf Unix-Systemen und Windows). Die Standardaus-/eingabe unterstützt jedoch in der Regel nicht alle auf dem jeweiligen System möglichen Dateiformate. Diese Einschränkung können Sie vermeiden, indem Sie anstelle der Standardaus-/eingabe die Variable %TEMPFILE verwenden. Dies hat den Vorteil, dass beliebige Dateiformate verwendet werden können. Auch wenn ein Vorverarbeitungskommando nicht auf Standardausgabe ausgeben, bzw. ein Nachverarbeitungskommando nicht von Standardeingabe lesen kann, hilft in der Regel die Angabe von %TEMPFILE in den Aufrufparametern.

Vorverarbeitung und Nachverarbeitung sind Teil der Auftragsklammer. Der Auftraggeber erhält also immer eine Rückmeldung über den erfolgreichen bzw. erfolglosen Abschluss der Vor-/Nachverarbeitung.

Folgendes ist bei Verwendung der Vor- und Nachverarbeitungsfunktion zu beachten:

- Die Vor-/Nachverarbeitung läuft als Teil der Dateiübertragung unter der selben Zugangsberechtigung ab wie diese. Diese Angaben stehen entweder explizit im Dateiübertragungsauftrag oder in der USER-ADMISSION eines Berechtigungsprofils.
- Wird der Auftrag über ein FTAC-Profil abgewickelt, so muss im Profil die Funktion FILE-PROCESSING zugelassen, oder ein Dateinamenspräfix mit Pipe-Zeichen '|' am Beginn festgelegt sein.
- Bei nicht wiederanlauffähigen Vor-/Nachverarbeitungen muss die Verbindung zum Partner bestehen bleiben, bis die gesamte Verarbeitung abgeschlossen ist.

### Wiederanlauffähigkeit bei Vor- und Nachverarbeitung

Bei wiederanlauffähiger Vor- und Nachverarbeitung werden die Daten zur Übergabe zwischen openFT und dem Verarbeitungskommando immer in einer temporären Datei abgelegt. Dadurch gliedert sich der Auftrag in 3 Phasen: Vorverarbeitung, Übertragung, Nachverarbeitung.

Die Wiederanlauffähigkeit einer Vor- oder Nachverarbeitung wird erreicht, indem Sie beim Übertragungskommando zusätzlich ein "&" vor dem Vor- bzw. Nachverarbeitungskommando angeben. Dabei verhalten sich Aufträge mit openFT-Partnern wie folgt:

- Verbindungsverlust während der Vorverarbeitung:
  Geht beim Ausführen des Vorverarbeitungskommandos die Verbindung verloren, dann wird das Kommando auch nach dem Verbindungsverlust noch vollständig ausgeführt. Bei einem Wiederanlauf nach Beendigung des Kommandos wird die temporäre Datei übertragen.
- Verbindungsverlust während der Übertragung:
  In diesem Fall führt openFT für die temporäre Datei wie üblich einen Wiederanlauf durch.
- Verbindungsverlust während der Nachverarbeitung:
  Geht die Verbindung während der Nachverarbeitung verloren, dann wird das Nachverarbeitungskommando noch vollständig ausgeführt. Bei einem Wiederanlauf werden dann alle restlichen, zum openFT-Auftrag gehörenden Aktionen durchgeführt (z.B. eine Folgeverarbeitung oder die Ergebnismeldung an den Partner).

*Hinweise für BS2000-Systeme*

Das openFT-Subsystem kann nicht beendet werden, so lange es noch wiederanlauffähige Aufträge gibt, deren Vor- oder Nachverarbeitung noch nicht abgeschlossen ist. Dauert dies zu lange, kann der Systemverwalter die zu den Aufträgen gehörenden Batch-Jobs abbrechen (CANCEL-Kommando). In diesem Fall wird beim Neustart von openFT die Meldung FTR2083 ausgegeben.

Sind beim Beenden des openFT mit STOP-FT noch wiederanlauffähige Aufträge aktiv, die sich in der Kommandoausführungsphase befinden, dann wird der Shutdown um bis zu 2 Minuten verzögert. Ist die Kommandoausführung nach 2 Minuten noch nicht abgeschlossen, wird openFT beendet, beim nächsten START-FT wird der Auftrag abgebrochen.

*Hinweise für Unix- und Windows-Systeme*

Die temporäre Datei wird im Verzeichnis *.openFTTmp* gespeichert und erst nach ordnungsgemäßer oder fehlerhafter Beendigung des Kommandos wieder gelöscht.

*.openFTTmp* wird von openFT neu angelegt, falls es noch nicht existiert. Es befindet sich im Home-Verzeichnis des jeweiligen Benutzers. Auf dem lokalen Rechner ist dies der Benutzer, unter dessen Kennung der Auftrag gestartet wurde. Auf dem fernen Rechner ist es der Benutzer, dessen Kennung angegeben wurde bzw. dem die angegebene Zugangsberechtigung gehört.

Wenn bei einem openFT-Shutdown noch wiederanlauffähige Aufträge aktiv sind und sich in der Kommandoausführungsphase befinden, wird der Shutdown um bis zu 10 Minuten verzögert, um die Kommandos noch fertig auszuführen. Ein Kommando zum Beenden des openFT-Servers bleibt solange "hängen", die Eingabeaufforderung erhält man erst, wenn der Serverprozess beendet ist.

#### Serverfunktion für ferne Kommandoausführung (ftexec)

Eine spezielle Form der Vorverarbeitung ist die Serverfunktion für die ferne Kommandoausführung (Kommando *ftexec* für Unix-/Windows-Systeme, EXECUTE-REMOTE-CMD für BS2000 und FTEXEC für z/OS ). Dieses Kommando ermöglicht es, auf einem fernen System Kommandos auszuführen. Der Exit-Code sowie die Ausgaben von *stdout* und *stderr* (Windows- und Unix-Systeme), SYSLST und SYSOUT ( BS2000) bzw. STDOUT= SYSPRINT und STDERR=SYSTSPRT (z/OS) werden am lokalen Rechner ausgegeben. Damit verhält sich dieses Kommando wie eine Ausführung der entsprechenden Kommandos am lokalen Rechner.

### 4.7.9 Folgeverarbeitung

openFT bietet vier verschiedene Arten von Folgeverarbeitungsaufträgen:

- Folgeverarbeitung im lokalen System nach erfolgreicher Dateiübertragung
- Folgeverarbeitung im fernen System nach erfolgreicher Dateiübertragung
- Folgeverarbeitung im lokalen System nach nicht erfolgreicher Dateiübertragung
- Folgeverarbeitung im fernen System nach nicht erfolgreicher Dateiübertragung

Für die Syntax und Bearbeitung der Anweisungen und Kommandos sind die Konventionen des Systems entscheidend, auf dem die Folgeverarbeitung ausgeführt werden soll. Eine Kommandofolge im fernen System kann nur dann verarbeitet werden, wenn im fernen System ein FT-Produkt eingesetzt wird, das diese Funktion unterstützt.

#### Variablen bei der Folgeverarbeitung

Innerhalb des Kommandos bzw. der Kommandofolge für die Folgeverarbeitung können Sie Variablen angeben, z.B. für den Partnernamen oder den Dateinamen. Diese werden beim Start der Folgeverarbeitung im jeweiligen System durch Werte ersetzt, die sich aus dem Dateiübertragungsauftrag ergeben.

Details mit den zulässigen Variablen finden Sie im jeweiligen openFT-Handbuch "Kommandoschnittstelle".

### Maximale Länge der Folgeverarbeitung

Sowohl für das lokale als auch für das ferne System dürfen die Angaben für die Folgeverarbeitung, abhängig von der eingesetzten openFT-Version, jeweils zusammen nicht mehr als maximal 1000 Zeichen betragen. Wieviele Zeichen ausgewertet werden, ist betriebssystemabhängig und kann den entsprechenden FT-Beschreibungen entnommen werden. Dabei ist zu beachten, dass

- die Längenbeschränkung nach einer eventuellen Variablenersetzung gilt.
- ein Folgeverarbeitungskommando in einem Windows-System ab openFT V12 in den Zeichencode UTF-8 umgewandelt wird, und dadurch ein Zeichen außerhalb des ISO646-Zeichenvorrats mehr als ein Byte Speicherplatz belegt.

Die Begrenzung von bis zu 1000 Zeichen kann umgangen werden, indem innerhalb der Folgeverarbeitung eine Prozedur, ein Shell-Skript oder ein Programm aufgerufen wird. Eine Prozedur kann die Kommandofolge enthalten, die bei erfolgreicher bzw. nicht erfolgreicher Dateiübertragung ausgeführt werden soll.

### Folgeverarbeitung bei FTP- und FTAM-Partnern

Bei Kopplung mit FTP- oder FTAM-Partnern gibt es Einschränkungen, da mit dem FTP- bzw. FTAM-Protokoll keine Folgeverarbeitungsdaten übertragen werden können. Eine Folgeverarbeitung im FTP- bzw. FTAM-Partnersystem ist nur möglich, wenn sie dort in einem FTAC-Berechtigungsprofil fest vorgegeben ist. Im lokalen System kann immer eine Folgeverarbeitung angestoßen werden.

### Sende- oder Empfangsdatei nach Übertragung löschen

Für Aufträge, bei denen die Sendedatei nach erfolgreicher Übertragung gelöscht werden soll, gibt es die spezielle Folgeverarbeitung *DELETE. Diese Funktion steht im lokalen System für alle Partner zur Verfügung. Im fernen System steht diese Funktion nur für openFT- und FTAM-Partner zur Verfügung.

Um bei einer misslungenen Übertragung nicht undefinierte Dateifragmente zu haben, ist es sinnvoll, nach einer misslungenen Übertragung über die Folgeverarbeitung die Empfangsdatei zu löschen.

## 4.8 Zugriffsschutz für Sende- und Empfangsdatei

openFT ohne die FTAC-Funktionalität bietet den gleichen Zugangs- und Zugriffsschutz wie das Betriebssystem. Der FT-Benutzer muss sich für den Zugriff auf eine Datei über ein FT-System ebenso mit Berechtigungen ausweisen, wie er sich gegenüber dem Datenverwaltungssystem des Betriebssystems ausweisen müsste. Das bedeutet, dass eine komplette LOGON-/login-Berechtigung sowie gegebenenfalls ein Dateikennwort benötigt wird.

Bei Einsatz von openFT mit FTAC-Funktionalität wird der auf Mitteln des Betriebssystems basierende Zugangs- und Zugriffsschutz um die in der FTAC-Funktionalität enthaltenen Sicherheitsmechanismen erweitert.

### 4.8.1 Zugriffsschutz während der Übertragung

Beachten Sie bitte, dass die Zieldatei in der Regel nicht während der ganzen Zeit der Auftragsbearbeitung gegen das Überschreiben durch andere Benutzer geschützt ist. Wird die Übertragung z.B. unterbrochen, dann können eventuell andere Benutzer schreibend auf die Zieldatei zugreifen. Der Zugriffsschutz ist bei den einzelnen Systemen unterschiedlich:

#### BS2000-Systeme

Bei openFT (BS2000) wird eine Dateisperre verwendet, die die Dateien auch bei Übertragungsunterbrechungen und zwischen Auftragsannahme und Auftragsabarbeitung schützt. Dieser Schutz gilt nicht für Bibliothekselemente und POSIX-Dateien.

Die Übertragungsdateien werden bei Annahme eines Auftrags mittels eines Dateilocks gesperrt. Sendedateien können dann von anderen Zugreifern noch gelesen werden, für Empfangsdateien ist jeder Zugriff verboten. Das BS2000-Kommando SHOW-FILE-LOCK zeigt, ob eine Datei durch openFT gesperrt ist, und listet die betroffenen Transfer-ID's auf. Der Dateilock wird beim Entladen des Subsystems entfernt.

Die Arbeitsweise von openFT bedingt, dass eine schon vorhandene Empfangsdatei nur überschrieben werden kann, wenn für diese Datei sowohl Lese- als auch Schreibberechtigung gegeben ist. Für den Dateizugriff müssen auch die Festlegungen der ACL (Access Control List) bzw. BASIC-ACL erfüllt sein.

Unter welchen Bedingungen der FT-Benutzer auf eine BS2000-Datei zugreifen kann, zeigt die folgende Tabelle:

| Zugriffsart | Bedingungen für den Dateizugriff |
|---|---|
| Lesen einer Sendedatei | – Datei unter angegebener Benutzerkennung katalogisiert oder<br>– Datei mehrfachbenutzbar oder<br>– Benutzer arbeitet unter der Kennung TSOS und<br>– lesender Zugriff erlaubt und<br>– gültiges Kennwort angegeben, wenn die zu übertragende Datei mit einem Lese- oder Execute-Kennwort geschützt ist |
| Überschreiben einer bereits vorhandenen Empfangsdatei | – Datei unter angegebener Benutzerkennung katalogisiert oder<br>– Datei mehrfachbenutzbar oder<br>– Benutzer arbeitet unter der Kennung TSOS und<br>– lesender und schreibender Zugriff erlaubt und<br>– gültiges Kennwort angegeben, wenn die Datei mit einem Kennwort geschützt ist |

### z/OS-Systeme

openFT (z/OS) sperrt die Sende- und Empfangsdatei nur dann gegen simultane (Schreib-)Zugriffe, wenn tatsächlich Daten übertragen werden, d.h wenn sich der Auftrag im Zustand ACTIVE befindet. Die Sende- und die Empfangsdatei sind demnach nicht gesperrt, wenn die Dateiübertragung noch nicht begonnen hat oder gerade unterbrochen ist.

Falls openFT auf eine Sende- oder Empfangsdatei zuzugreifen versucht, die gesperrt ist (z.B. weil schon ein anderer FT-Auftrag darauf zugreift), dann wird der FT-Auftrag abgelehnt oder abgebrochen.

Bei einem Member eines PO- oder PDSE-Datasets bedeutet das:

- Wenn ein Member eines PO- oder PDSE-Datasets gelesen werden soll (Sendedatei), so darf zum Zeitpunkt der Auftragserteilung kein anderes Member desselben Datasets anderweitig zum Schreiben oder exklusiv zum Lesen geöffnet sein oder vor dem Ende der Dateiübertragung geöffnet werden.
- Wenn ein Member eines PO- oder PDSE-Datasets geschrieben werden soll (Empfangsdatei), so darf weder ein anderes Member desselben Datasets noch das Dataset selbst zum Zeitpunkt der Auftragserteilung anderweitig geöffnet sein oder vor dem Ende der Dateiübertragung geöffnet werden. In diesem Fall (Empfangsdatei) führt unter Umständen schon das Anzeigen der Member-Liste (z.B. beim Anstoßen eines Sendeauftrags über die Menüschnittstelle, siehe Handbuch "openFT (z/OS) - Kommandoschnittstelle", oder allgemein die Verwendung der PDF-Funktion "member list") zum Abbruch des FT-Auftrags.

Beim Abbruch eines FT-Auftrags wegen Zugriffsversuchs auf eine gesperrte Datei wird eine Fehlermeldung ausgegeben.

**Andere Systeme**

In anderen Systemen, beispielsweise Windows- und Unix-Systemen oder auch im BS2000, wenn es sich um POSIX-Dateien oder Bibliothekselemente handelt, muss der Benutzer selbst den exklusiven Zugriff auf seine zu übertragenden Dateien garantieren. In diesen Systemen kann auch während der Übertragung die Datei nicht exklusiv von openFT belegt werden.

Der Benutzer muss daher selbst sicherstellen, dass die zu übertragende Datei während der gesamten Dauer des FT-Auftrags konsistent ist (Daten und Attribute der Datei). Das gilt sowohl für die Sendedatei als auch für die Empfangsdatei. Die Gefahr eventueller Inkonsistenzen durch Mehrfachzugriffe kann beispielsweise durch Zugriffseinschränkungen verringert werden (Kommando *chmod* in Unix-Systemen). Außerdem besteht die Möglichkeit, die Datei auf einen anderen Namen bzw. in ein temporäres Verzeichnis zu transferieren und erst nach erfolgreicher Dateiübertragung mit Hilfe der Folgeverarbeitung umzubenennen bzw. in ein anderes Verzeichnis zu verschieben.

### 4.8.2 Besonderheiten in z/OS

Die im z/OS-System installierten Software-Produkte SYS1.UADS und RACF (oder dazu kompatible Produkte wie TOP-SECRET und ACF-2) werden für die Prüfung der Zugangs- und Zugriffsberechtigung des FT-Benutzers herangezogen. Daher gelten beim lesenden und schreibenden Dateizugriff gleiche Bedingungen für openFT- und TSO- bzw. JES2-/JES3-Benutzer.

Bei Sendedateien und bei bereits existierenden Empfangsdateien prüft openFT die Zugriffsberechtigung (lesen/schreiben) des FT-Benutzers mit Hilfe der oben genannten Produkte anhand von User-Id und Password, die in der TRANSFER-ADMISSION angegeben sind, sowie gegebenenfalls anhand des Datei-Passwords. Wenn diese Prüfung nicht erfolgreich ist, dann wird der File Transfer nicht durchgeführt; der Benutzer erhält eine entsprechende Meldung.

Aus Datenschutzgründen wird bei dieser Meldung nicht mitgeteilt, welcher der Parameter USER-IDENTIFICATION, ACCOUNT oder PASSWORD bzw. Datei-Passwort betroffen ist.

Wenn eine Empfangsdatei noch nicht existiert, dann wird sie von openFT eingerichtet. Auch dabei wird von openFT über die oben genannten Produkte anhand von User-Id und Password die Zugriffsberechtigung (schreiben) geprüft. Wenn diese Prüfung nicht erfolgreich ist, dann wird der File Transfer nicht durchgeführt und die gleiche Meldung wie oben ausgegeben.

openFT vergibt für neu eingerichtete Dateien aktiv keine Zugriffsschutz-Merkmale. Insbesondere wird weder ein Datei-Passwort an das System übergeben noch wird das so genannte "RACF-Bit" im DSCB von openFT gesetzt. Falls eine neu eingerichtete Datei unter z/OS sofort gegen unberechtigte Zugriffe geschützt sein soll, wird die Verwendung der nachfolgend näher beschriebenen RACF-Funktion "generic profile" empfohlen.

Mit Hilfe dieser RACF-Funktion kann eine Gruppe von Dateien, deren Namen ähnlich aufgebaut sind, mit einem gemeinsamen Schutz versehen werden. Zum Beispiel können alle Dateien einer bestimmten User-Id, deren Namen die Zeichenfolge TRANS enthalten, gegen Zugriffe von fremden User-Id's geschützt werden. Dies gilt auch für Dateien, die von openFT neu angelegt werden.

Dateischutzmerkmale der Sendedatei werden nicht in das Empfangssystem übertragen und können daher auch nicht für eine neu eingerichtete Empfangsdatei übernommen werden.

openFT setzt keine Veränderungsschutzfrist für die Datei.

## 4.9 Dateimanagement

Das Dateimanagement von openFT ist sowohl im fernen als auch im lokalen System möglich.

Ist das Partnersystem ein z/OS-System, dann gelten einige Besonderheiten, z.B. steht auf z/OS kein lokales Dateimanagement zur Verfügung. Details finden Sie im Handbuch "openFT (z/OS) - Kommandoschnittstelle".

### 4.9.1 Dateimanagement im fernen System

openFT bietet die Möglichkeit, Dateien im fernen System vom lokalen System aus zu verwalten (Dateimanagement). Sie können im Partnersystem

- Inhalte von Dateiverzeichnissen auflisten,
- Dateiattribute z.B. die Größe einer Sendedatei, abfragen,
- Attribute von Dateien ändern, z.B. Dateien umbenennen,
- Dateien löschen,
- Verzeichnisse anlegen, umbenennen und löschen.

Partnersysteme, die die Dateimanagement-Funktionen unterstützen, können ebenfalls die Initiative für solche Aufträge ergreifen und vom fernen System aus auf Ihr lokales System entsprechend zugreifen. In beiden Fällen schickt das System, in dem die Initiative ergriffen wird, eine Beschreibung des Auftrags an das Partnersystem. Das Partnersystem führt den Auftrag seinen Konventionen entsprechend aus.

Die Dateimanagement-Funktionen werden über das jeweils verwendete Protokoll erbracht (openFT, FTAM oder FTP). Protokollbedingte Unterschiede zwischen openFT-, FTP- und FTAM-Partnersystemen gibt es beim Ändern der Dateiattribute. Abhängig vom Protokoll lassen sich folgende Attribute einer Datei ändern, sofern es das ferne System unterstützt:

| **Attribut** | **FTAM-Partner** | **openFT-Partner** | **FTP-Partner** |
|---|---|---|---|
| Dateiname (FILE-NAME/NEW-NAME) | X | X | X |
| Zugriffsrechte (ACCESS-MODE) | X | X | |
| Verfügbarkeit der Datei (FILE-AVAILABILITY) | X | | |
| Abrechnungskonto für Dateispeicherungskosten (STORAGE-ACCOUNT) | X | | |
| rechtliche Bestimmungen bezüglich der Verwendung einer Datei (LEGAL-QUALIFICATION) | X | | |
| zukünftige Dateigröße (FUTURE-FILE-SIZE) | X | | |

### 4.9.2 Dateimanagement im lokalen System

Bei Einsatz der FTAM-Funktionalität haben Sie für die Kommunikation mit FTAM-Partnern die Möglichkeit, im lokalen System Dateien mit speziellen FTAM-Attributen zu versehen (siehe Seite 107). Die in diesem Zusammenhang angebotene Funktionalität ermöglicht Ihnen das Anzeigen und Ändern der FTAM-Attribute einer Datei im lokalen System.

Die FTAM-Attribute existieren nur im virtuellen Dateispeicher und sind in erster Linie für die Dateiübertragung und das Dateimanagement mit FTAM-Partnern gültig. Im lokalen System bleibt die betriebssystemspezifische Einstellung der Dateiattribute unverändert. Das heißt, dass auch weiterhin mit betriebssystemspezifischen Kommandos Dateien bzw. Dateimerkmale verändert werden können. Beispielsweise kann eine Datei mit einem systemspezifischen Löschkommando gelöscht werden, obwohl eine entsprechende Einstellung von PERMITTED-ACTION das Löschen dieser Datei für FTAM-Partner untersagt.

Die folgende Tabelle zeigt die Dateimanagement-Funktionen im lokalen System:

| **FTAM-Attribut** | **anzeigen[1]** | **ändern** |
|---|---|---|
| Dateiname (FILE-NAME) * | X | |
| Abrechnungsnummer (STORAGE-ACCOUNT) | X | |
| Art der letzten Dateinutzung * | X | |
| Name des letzten Dateibenutzers * | X | |
| Datum und Uhrzeit der letzten Veränderung des Dateiinhalts | X | |
| Dateityp (DATA-TYPE) | X | X |
| Zeichensatz (CHARACTER-SET) * | X | X |
| Satzformat (RECORD-FORMAT) * | X | X |
| maximale Satzlänge (RECORD-SIZE) * | X | X |
| Verfügbarkeit der Datei (FILE-AVAILABILITY) * | X | |
| Dateizugriffsrechte (PERMITTED-ACTIONS) * | X | X |
| aktuelle Größe der Datei in Byte (CURRENT-FILE-SIZE) * | X | |
| mögliche Größe der Datei in Byte (FUTURE-FILE-SIZE) | X | |
| rechtliche Bestimmungen (LEGAL-QUALIFICATION) | X | |

[1] Bei lokalem Dateimanagement werden nur die mit Stern (*) gekennzeichneten FTAM-Attribute angezeigt, bei fernem Dateimanagement alle Attribute.

Folgende FTAM-Attribute werden auch bei Dateiübertragungen mit dem openFT-Protokoll und teilweise dem FTP-Protokoll ausgewertet:

- Dateityp (DATA-TYPE)
- Satzformat (RECORD-FORMAT)
- maximale Satzlänge (RECORD-SIZE)

Wenn sich die im Dateiübertragungsauftrag angegebenen Formatattribute und diese FTAM-Attribute widersprechen, wird der Auftrag in der Regel abgelehnt. Um dies zu vermeiden, können die FTAM-Attribute in der lokalen Datei gelöscht werden, ohne die Datei selbst zu löschen.

Diese FTAM-Attribute werden nur bei einem Dateiübertragungsauftrag per FTAM-Protokoll gesetzt (nicht bei einem Auftrag per openFT- oder FTP-Protokoll).

## 4.10 Besonderheiten beim File Transfer mit FTAM-Partnern

Die FTAM-Funktionalität erlaubt es Ihnen, auf Basis des ISO-Protokolls ISO8571 File Transfer zu betreiben. Die folgenden Abschnitte beschreiben für "FTAM-Spezialisten" die Besonderheiten hinsichtlich Übertragung und Abbildung FTAM-spezifischer Dateiattribute beim File Transfer mit FTAM-Partnern.

Das FTAM-Protokoll wird auf BS2000-, Unix- und Windows-Systemen unterstützt.

### 4.10.1 Virtueller Dateispeicher

Jedes System, das den File Transfer über FTAM-Protokolle ermöglicht, muss seine Dateien dem Partner in einer normgemäßen Beschreibung (ISO8571) zur Verfügung stellen. Zu diesem Zweck werden die Attribute einer Datei vom realen Dateispeicher auf einen virtuellen Dateispeicher abgebildet und umgekehrt. Der virtuelle Dateispeicher hat demzufolge keinen Einfluss auf die Dateimerkmale der Dateien im lokalen System, sondern hat nur die Aufgabe, Dateimerkmale ins ferne FTAM-System zu transportieren. Im Folgenden werden die Beschreibungskriterien einer Datei im virtuellen Dateispeicher vorgestellt. Die Darstellung im virtuellen Dateispeicher ist durch die FTAM-Norm vorgegeben. Dazu werden im wesentlichen drei Gruppen von Dateiattributen unterschieden:

Kernel group

beschreibt die wesentlichen Attribute der Dateien. Diese werden bei der Dateierzeugung festgelegt. Sie enthalten z.B. den Dateinamen, Informationen über Dateistruktur und -inhalt und geben Auskunft über die vereinbarten Dateizugriffsrechte.

Storage group

umfasst die Speicherattribute von Dateien. Zu den Speicherattributen zählen u.a. Dateigröße, Dateiverfügbarkeit, Datum und Uhrzeit der Dateierzeugung, des letzten lesenden bzw. schreibenden Zugriffs oder der letzten Änderung von Dateiattributen sowie die Identifikation des Benutzers, der diese Änderungen und Zugriffe verursacht hat.

Security group

definiert Sicherheitsattribute bezüglich Zugang und Zugriff.

#### Attribute der Kernel group

Die Kernattribute der Kernel group werden bei der Erzeugung der Datei gesetzt und machen grundsätzliche Angaben zur Datei:

filename
: enthält den Dateinamen.

permitted actions

legt fest, welche Aktionen mit einer Datei durchgeführt werden können:

- Datei lesen (READ-FILE)
- Dateneinheit einfügen (INSERT-DATA-UNIT)
- Datei ersetzen (REPLACE-FILE)
- Datei erweitern (EXTEND-FILE)
- Dateneinheit löschen (ERASE-DATA-UNIT)
- Dateiattribute lesen (READ-ATTRIBUTES)
- Dateiattribute ändern (CHANGE-ATTRIBUTES)
- Datei löschen (DELETE-FILE)

Außerdem definieren die *permitted actions* die Methode, mit der auf strukturierte Dateien zugegriffen werden kann (siehe auch Abschnitt „FTAM-Dateien“ auf Seite 73)

- vorwärts (TRAVERSAL)
- rückwärts (REVERSE TRAVERSAL)
- beliebig (RANDOM)

contents type

Definiert die Datenstruktur und die Methode, mit der auf die strukturierten Daten zugegriffen werden kann.

### Attribute der Storage group

Die Speicherattribute der Storage group beschreiben die Speichereigenschaften der Datei, beispielsweise wer wann wie zuletzt auf die Datei zugegriffen hat. Einige dieser Eigenschaften werden automatisch geändert, wenn die Datei gelesen oder verändert wird. Sie können aber nicht direkt durch Benutzerkommandos verändert werden. Auf direkt veränderbare Attribute können Sie mit openFT Einfluss nehmen.

| **Attribut**[1] | **Definition** |
|---|---|
| storage account * | nennt die Stelle, die für die Speicherkosten der Datei zuständig ist |
| date and time of creation | Datum und Uhrzeit der Erzeugung |
| date and time of last modification | Datum und Uhrzeit der letzten Änderung |
| date and time of last read access | Datum und Uhrzeit des letzten Lesezugriffs |
| date and time of last attribute modification | Datum und Uhrzeit der letzten Änderung der Attribute |
| identity of creator | Benutzer, der die Datei erzeugt hat. |
| identity of last modifier | Benutzer, der die Datei zuletzt geändert hat. |
| identity of last reader | Benutzer, der die Datei zuletzt gelesen hat. |

| Attribut[1] | Definition |
|---|---|
| identity of last attribute modifier | Benutzer, der zuletzt die Attribute der Datei geändert hat. |
| file availability * | Information, ob eine Datei sofort verfügbar ist oder erst irgendwoher, z.B. aus einem Archiv, beschafft werden muss. |
| filesize | beschreibt die Speicherkapazität, die im realen Dateispeicher beansprucht wird. Deshalb kann eine Datei in Systemen mit unterschiedlicher Darstellung der Dateitypen auch unterschiedliche Dateigrößen haben. Einige Dateispeicher weisen für die Speicherung von Dateien ein Vielfaches einer Basiseinheit, zum Beispiel Blöcke, zu. *filesize* gibt dann einen Wert an, der nicht der Dateigröße entspricht. |
| future filesize * | beschreibt die zukünftige Dateigröße. Darunter ist die Größe der Datei zu verstehen, die die Datei nach Bearbeitung annehmen kann. Der Initiator kann den Wert von *future filesize* ändern. Sobald die Datei den angegebenen Wert für die Dateigröße erreicht hat, kann der Responder den Wert mit oder ohne Warnung an den Initiator erhöhen oder die Änderung des Wertes mit entsprechender Fehlermeldung ablehnen. |

[1] Mit Stern (*) gekennzeichnete Attribute sind direkt veränderbar.

### Attribute der Security group

Das FTAM-Konzept des virtuellen Dateispeichers sieht für den Zugriffsschutz die Security group vor.

| Attribut[1] | Definition |
|---|---|
| access control * | Bedingungen, unter denen der Zugriff auf Dateien erlaubt ist. Beispiele hierfür sind Kennwörter für verschiedene Zugriffsarten (lesen, einfügen, ersetzen, vergrößern) oder Sperren, mit denen der gleichzeitige Dateizugriff durch verschiedene Benutzer geregelt wird. |
| legal qualifications * | Angabe zum rechtlichen Status der Datei und ihrer Verwendung. Es gibt derzeit keine anerkannte Interpretation dieses Attributs, das heißt, es ist vom jeweiligen Partner abhängig, wie dieses Attribut interpretiert wird. |

[1] Mit Stern (*) gekennzeichnete Attribute sind direkt veränderbar.

Die Abbildung der Dateizugriffsrechte und der FTAM-Dateiattribute auf das reale Dateisystem ist im Handbuch "openFT (BS2000) - Installation und Betrieb" bzw. Handbuch "openFT (Unix- und Windows-Systeme) - Installation und Betrieb" beschrieben.

## 4.10.2 Adressierung über Application Entity Title (AET)

In der OSI-Welt werden die Kommunikationspartner durch Anwendungsinstanzen (Application Entities) repräsentiert. Eine Anwendungsinstanz ist eine adressierbare Einheit in Schicht 7 des OSI-Referenzmodells (Anwendungsschicht). Eine solche Anwendungsinstanz ist z.B. der Zugriffspunkt (Access Point) einer FTAM-Anwendung, über den sich ein OSI-TP-Kommunikationspartner an die FTAM-Anwendung binden kann. In der OSI-TP-Norm wird jeder Anwendungsinstanz ein Application Entity Title zugeordnet, über den die Anwendungsinstanz im OSI-Netz eindeutig adressierbar ist.

In der ISO-Norm sind zwei Formen des AET definiert, die Directory-Form (transparentes Format) und die Object-Identifier-Form (numerisches Format).

openFT (Unix-Systeme) und openFT (Windows) unterstützen mit ihrer FTAM-Funktionalität beide Formen des AET.

Ein AET besteht aus zwei Teilen:

– Application Process Title (APT)
– Application Entity Qualifier (AEQ).

Bei der Übertragung mit dem FTAM-Protokoll schickt openFT standardmäßig einen Nil-Application Entity Title als calling bzw. called Application Entity Title mit. In Unix- und Windows-Systemen kann dieses Verhalten per Betriebsparameter geändert werden. Auf BS2000-Systemen kann das Senden von Nil-AET's über einen optionalen Rep systemglobal eingestellt werden.

Der Nil-AET lautet:
`1.3.9999.1.7`

### 4.10.2.1 FTAM-Partner mit AET im Object-Identifier-Format adressieren

Wenn ein called AET vom "Nil-Application Entity Title" abweichen soll, dann muss er in der Partnerliste bei der Instanzidentifikation angegeben werden.

Die Angabe hat folgende Syntax:

n1.n2[.n3] [.n10][..m]

n1.n2[.n3] [.n10]
: gibt den *application process title* an, zwischen zwei und zehn Dezimalzahlen, die jeweils durch einen Punkt getrennt sind. Der Wertebereich und die Bedeutung der Ziffern werden unten erklärt.

[..m] gibt den *application entity qualifier* an; Wertebereich siehe unten. Wird ein AEQ angegeben, müssen zwei Punkte davor stehen.

*Beispiel*

Ein FTAM-Partner auf dem Rechner *daisy2* mit APT=*1.0.56.881.4* und AEQ=*785* soll mit dem Namen *daisyftm* in die Partnerliste eingetragen werden. Dazu geben Sie folgendes Kommando:

Unix- und Windows-Systeme:

```
ftaddptn daisyftm -pa=ftam://daisy2 -id=1.0.56.881.4..785
```

BS2000-Systeme:

```
ADD-FT-PARTNER PARTNER-NAME=DAISYFTM, -
               PARTNER-ADDRESS=FTAM://DAISY2, -
               IDENTIFICATION=1.0.56.881.4..785
```

### Application Process Title (APT)

Der APT wird zur Kennzeichnung der Anwendung verwendet. Der APT sollte gemäß der OSI-Norm global, d.h. weltweit eindeutig sein. Aus diesem Grund sollte er von einem Standardisierungsgremium vergeben und registriert werden, z.B. in Deutschland von der DIN CERTCO Gesellschaft für Konformitätsbewertung mbH.

Ein APT in Object-Identifier-Form besteht aus maximal 10 Komponenten:
`(komponente1,komponente2,...,komponente10)`

Die Werte für komponente1 bis komponente10 sind bereits teilweise standardisiert. Hierbei wurde einigen Zahlen ein symbolischer Name zugeordnet. Der Wertebereich für komponente2 hängt vom Wert für komponente1 ab. In der folgenden Tabelle sind die durch die FTAM-Funktionalität unterstützten symbolischen Namen und die Wertebereiche dargestellt:

| **komponente1** | **komponente2** | **komponente3 bis komponente10** |
|---|---|---|
| 0: CCITT | 0: RECOMMENDATION<br>1: QUESTION<br>2: ADMINISTRATION<br>3: NETWORK-OPERATOR<br>(erlaubte Werte 0 - 39) | erlaubte Werte:<br>0 - 67 108 863 |
| 1: ISO | 0: STANDARD<br>1: REGISTRATION-AUTHORITY<br>2: MEMBER-BODY<br>3: IDENTIFIED-ORGANIZATION<br>(erlaubte Werte 0 - 39) | erlaubte Werte:<br>0 - 67 108 863 |
| 2:JOINT-ISO-CCITT | erlaubte Werte:<br>0 - 67 108 863 | erlaubte Werte:<br>0 - 67 108 863 |

Der APT, den Sie angeben, muss nicht von einem Standardisierungsgremium vergeben werden, d.h. Sie können den APT selbst vergeben. Er muss die beiden folgenden Anforderungen erfüllen:

– er muss innerhalb des Netzes eindeutig sein
– er muss aus Werten bestehen, die gemäß der obigen Tabelle zulässig sind

Ein ferner Partner, der AETs verlangt, muss diesen APT kennen, um eine Verbindung aufbauen zu können.

#### Application Entity Qualifier (AEQ)

Der AEQ identifiziert einen Zugriffspunkt innerhalb einer Anwendung. Den Zugriffspunkten einer Anwendung können Sie nur dann AEQs zuordnen, wenn Sie der Anwendung selbst einen APT zugeordnet haben. Er wird vom Anwendungsbetreiber vergeben.

Der AEQ ist eine positive ganze Zahl zwischen 0 und 67108863.

Denselben AEQ dürfen Sie innerhalb einer Anwendung nicht mehrfach verwenden, d.h. in einer Anwendung dürfen nie zwei Zugriffspunkte mit demselben AEQ existieren. Sie müssen jedoch nicht allen Zugriffspunkten in einer Anwendung einen AEQ zuordnen.

### 4.10.2.2 Erweiterte Unterstützung des Application Entity Title auf Windows- und Unix-Systemen

Die Ermittlung des Absenders beim FTAM-Protokoll erfolgte bislang ausschließlich über die Partneradresse. Wenn nun die Partneradresse des Initiators sich von Mal zu Mal ändert, kann es bei FTAM Probleme mit der Partnerzuordnung und auch mit dem Wiederanlauf geben.

Der Application Entity Title bietet nun die Möglichkeit, eine von der Partneradresse unabhängige Identifizierung eines Initiators auch für FTAM zu realisieren. Per Standardeinstellungen in den Betriebsparametern kann die Absenderüberprüfung von FTAM-Partnern mittels Application Entity Title eingestellt werden.

#### Directory-Format des Application Entity Title

Neben dem numerischen Format gibt es auch jeweils ein Directory-Format (auch transparentes Format genannt) für Application Process Title und Application Entity Qualifier. Eine im FTAM-Partnerlisteneintrag angegebene Instanzidentifikation wird als Directory-Format interpretiert, wenn sie nicht dem numerischen Format entspricht und Folgendes gilt:

– kein Leerzeichen an den Beginn einer Angabe

– Kein Leerzeichen an den Anfang eines Strings, der an eine numerische Angabe angehängt wird

Andernfalls wird der Application Entity Title abgelehnt.

Angaben, die ein Zeichen enthalten, das weder eine Dezimalziffer noch ein Punkt ist, werden als ISO646-String im Directory-Format des Application Entity Title gesendet bzw. ausgewertet.

Die Zeichenfolge bis zum letzten Auftreten zweier aufeinanderfolgender Punkte im Directory-Format wird als Application Process Title ausgewertet.

Die Zeichenfolge hinter der letzten Folge aus zwei Punkten wird als Application Entity Qualifier im Directory-Format interpretiert. Ausnahme: die Identifikation endet mit .. oder ..#.

**Beispiele**

Angaben im Directory-Format sind hexadezimal dargestellt, wie sie auch im Trace erscheinen würden:

| | |
|---|---|
| -id=1.3.5.6..# | APT = 0x312e332e352e36 *) |
| -id=9.3.4.5.2 | wird abgelehnt, da an erster Stelle nur 0, 1 oder 2 erlaubt wäre |
| -id=1.2.3.4.5.6.7..8.9 | APT = 0x312e322e332e342e352e362e37 AEQ =0x382e39 |
| -id=1.2.3.4.5.6.7.8.9.10.11 | wird abgelehnt |
| -id=%ip139.28.87.55 | APT = 0x2569703133392e32382e38372e3535 |
| -id=emil | APT = 0x656d696c |
| -id=1.3.5A | APT = 0x312e332e3541 |
| "-id=1.3.5 A" | wird abgelehnt |
| "-id=Emil Huber" | APT = 0x456d696c204875626572 |
| -id=Emil.Huber | APT = 0x456d696c2e4875626572 |
| -id=Emil..Huber | APT =0x456d696c AEQ = 0x4875626572 |
| -id=Emil.. | APT = 0x456d696c *) |
| -id=Emil..# | APT = 0x456d696c *) |

*) Diese Angaben sind nicht für eine Absenderüberprüfung per Application Entity Title geeignet, wenn der Partner openFT ist. D.h. führende Nullen sowie mit . oder .. endende Strings sind in numerischen Angaben zu vermeiden, und auch beim Directory-Format müssen .. oder ..# am Ende weggelassen werden.

### 4.10.3 Synchrone Übertragung mehrere Dateien mit FTAM

Mit einem *ncopy*-Kommando auf (Unix- und Windows-Systeme) und FTSCOPY auf (BS2000-Systeme) kann man mehrere Dateien über eine Transportverbindung und eine FTAM-Verbindung übertragen. Dies funktioniert für Sende- und Empfangsaufträge und wird durch die Dateinamen-Syntax gesteuert.

Die Gesamtlänge der Dateiliste muss in den Dateinamens-Parameter des ncopy- bzw. FTSCOPY-Kommandos passen.

Die Funktion steht auch an der Programmschnittstelle für Unix- und Windows-Systeme zur Verfügung.

#### Angabe der Sendedateien

Beginnt der Name der Sendedatei nicht mit zwei Kommas, so läuft alles wie bisher.

Beginnt der Name mit zwei Kommas, dann gilt:

- Aus diesem Namen werden mehrere Teilnamen abgeleitet, die wiederum durch jeweils zwei Kommas voneinander getrennt sind.

  *Beispiel 1*

  Name der Sendedatei im *ncopy*-/FTSCOPY-Kommando:

  ```
  ,,Brief1,,Dokument,,Buchung
  ```

  Es werden die drei Dateien *Brief1*, *Dokument* und *Buchung* übertragen.

- Endet der erste Teilname mit Schrägstrich, Punkt, oder in Windows auch Gegenschrägstrich, so wird er als Präfix für alle folgenden Teilnamen interpretiert.

  *Beispiel 2*

  Name der Sendedatei im *ncopy*-/FTSCOPY-Kommando:

  ```
  ,,MeinVerzeichnis/,,Brief1,,Dokument,,Buchung
  ```

  Es werden die Dateien *MeinVerzeichnis/Brief*, *MeinVerzeichnis/Dokument* und *MeinVerzeichnis/Buchung* übertragen.

Dateien mit einem Komma am Ende des Dateinamens oder mehreren Kommas innerhalb des Namens können mit der Mehrfach-Option nicht übertragen werden. Dateien, deren Namen mit zwei Kommas beginnen, können nur asynchron übertragen werden.

### Angabe der Empfangsdatei

Für die Einzel-Transfers bildet sich der Name der Empfangsdatei aus dem angegebenen Namen und dem Namen der Sendedatei, wobei ein evtl. angegebener Präfix im Sendedateinamen nicht berücksichtigt wird.

*Beispiel*

Wird in Beispiel 1 und Beispiel 2 *ziel/* als Empfangsdatei angegeben, so werden in beiden Fällen die Empfangsdateien *ziel/Brief1*, *ziel/Dokument* und *ziel/Buchung* erzeugt.

### Logging und Verhalten bei Übertragungsfehlern

Zu jeder Teilübertragung wird ein eigener Logging-Satz geschrieben, auch Folgeverarbeitungen (falls angegeben) werden für jede Teilübertragung einzeln gestartet.

Als Ergebnis einer derartigen Mehrfach-Übertragung kommt nur dann eine OK-Meldung, wenn alle Übertragungen erfolgreich waren. Ansonsten wird beim ersten Fehler abgebrochen und die entsprechende Meldung ausgegeben, auch wenn zuvor schon Dateien erfolgreich übertragen worden sind. Dateien, deren Übertragung noch nicht gestartet wurde, werden im Logging als "cancelled" markiert.

# 5 Sicherheitsfunktionen

openFT bietet folgende Sicherheitsfunktionen:

- FTAC-Funktionen
- Authentifizierung
- Erweiterte Absenderüberprüfung
- Verschlüsselung bei der Datenübertragung
- Schutzmechanismen gegen Datenmanipulation
- Verschlüsselung mit FTPS

## 5.1 FTAC-Funktionen

Die FTAC-Funktionalität bietet die Möglichkeit, FT-Aktivitäten durch Berechtigungssätze und Berechtigungsprofile rechner- und nutzerspezifisch zu steuern und zu kontrollieren:

- Im Berechtigungssatz werden für eine Benutzerkennung Sicherheitsstufen festgelegt (0 bis 100). Die Sicherheitsstufen haben Einfluss darauf, mit welchen Partnersystemen diese Kennung welche FT-Funktionen nutzen darf.
- Berechtigungsprofile definieren eine Zugangsberechtigung, die in FT-Aufträgen statt der LOGIN- oder LOGON-Berechtigung angegeben werden muss. Zugriffsrechte auf eine Benutzerkennung werden darin festgelegt, indem die Verwendung von Parametern in FT-Aufträgen eingeschränkt wird.
- Der FT-Verwalter muss bei Einsatz von FTAC den Partnersystemen Sicherheitsstufen zuordnen, siehe Abschnitt „FTAC-Sicherheitsstufen für Partner in der Partnerliste“ auf Seite 55).

**Warnung!**
Es ist zu beachten, dass openFT-AC nur für angeschlossene Produkte wie openFT wirksam ist. Wenn also im System weitere Dateitransferprodukte ohne openFT-AC Anschluss eingesetzt werden, ist ein abgestimmtes Sicherheitskonzept sinnvoll.

### 5.1.1 Berechtigungssätze

Berechtigungssätze realisieren die Sicherheitseinstellung anhand der Grundfunktionen von openFT. Die Verwaltung von Berechtigungssätzen ist in erster Linie Aufgabe des FTAC-Verwalters.

Es gibt:

- den systemweit gültigen Standardberechtigungssatz
- individuelle Berechtigungssätze für einzelne Benutzerkennungen

#### Standardberechtigungssatz

Die Vorgaben des Standardberechtigungssatzes gelten für alle Benutzerkennungen. Daher darf nur der FTAC-Verwalter den Standardberechtigungssatz ändern.

Nach der Installation steht der Standardberechtigungssatz auf BS2000- und z/OS-Systemen auf 0, d.h. es ist kein File Transfer möglich. Auf Unix- und Windows-Systemen steht der Standardberechtigungssatz auf 100, d.h. File Transfer ist uneingeschränkt möglich.

#### Individuelle Berechtigungssätze

Im Berechtigungssatz wird für jede der sechs Grundfunktionen (Inbound Empfangen und Senden, Inbound Folgeverarbeitung, Inbound Dateimanagement, Outbound Empfangen und Senden) eine maximale Sicherheitsstufe angegeben.

Ein Benutzer darf nur seinen eigenen Berechtigungssatz ändern, der FTAC-Verwalter darf die Berechtigungssätze aller Benutzer ändern. Daher gibt es für diese Grundfunktionen eine Vorgabe des FTAC-Verwalters (so genannte ADMIN LEVELS) und die Vorgabe des betreffenden Benutzers (USER LEVELS). Daraus ergeben sich für jede Grundfunktion folgende Möglichkeiten:

- Haben weder der betreffende Benutzer noch der FTAC-Verwalter eine Einstellung geändert, dann gilt die Einstellung des Standardberechtigungssatzes.
- Hat entweder der Benutzer oder der FTAC-Verwalter die Sicherheitsstufe geändert, dann gilt diese Einstellung. Eine vom Benutzer geänderte Sicherheitsstufe darf nur gleich oder niedriger sein als die des Standard-Berechtigungssatzes. Bei einer höheren Sicherheitsstufe wird eine Warnung ausgegeben.
- Haben beide die Einstellung geändert, dann gilt die niedrigere Sicherheitsstufe von beiden.

*Beispiel*

Die Spalte *Gültige Stufe* zeigt, wie sich die Einstellungen von Benutzer, FTAC-Verwalter und Standard-Berechtigungssatz auswirken.

| **Grundfunktion** | **Benutzer** | **FTAC-Verwalter** | **Standard-Berechtigungssatz** | **Gültige Stufe** |
|---|---|---|---|---|
| Outbound Senden | -- | -- | 100 | 100 |
| Outbound Empfangen | -- | 90 | 100 | 90 |
| Inbound Senden | 50 | -- | 90 | 50 |
| Inbound Empfangen | 50 | 10 | 90 | 10 |
| Inbound Verarbeitung | 50 | 10 | 50 | 10 |
| Inbound Datei-management | 10 | 20 | 50 | 10 |

-- bedeutet, dass keine Einstellung vorgenommen wurde

Die Benutzerkennung, zu der der Berechtigungssatz gehört, kann die Grundfunktion dann mit allen Partnersystemen nutzen, die höchstens diese Sicherheitsstufe haben, d.h. die Berechtigung wird bei einem openFT-Auftrag (Outbound und Inbound) mit der FTAC-Sicherheitsstufe des jeweiligen Partners verglichen, siehe auch Seite 55.

*Beispiel*

Die Spalte *Wirkung* zeigt für die Partner FT1, FT2 und FT3, wie sich die Einstellungen im Berechtigungssatz und in der Partnerstufe auswirken.

| **Grundfunktion** | **Stufe im Berechtigungsatz** | **Partnerstufe** | | | **Wirkung** | | |
|---|---|---|---|---|---|---|---|
| | | FT1 | FT2 | FT3 | FT1 | FT2 | FT3 |
| Outbound Senden | 100 | 100 | 90 | 10 | + | + | + |
| Outbound Empfangen | 90 | | | | - | + | + |
| Inbound Senden | 50 | | | | - | - | + |
| Inbound Empfangen | 10 | | | | - | - | + |
| Inbound Verarbeitung | 10 | | | | - | - | + |
| Inbound Datei-management | 10 | | | | - | - | + |

+ Auftrag wird akzeptiert
- Auftrag wird abgelehnt

### 5.1.2 Berechtigungsprofile

Ein Berechtigungsprofil ist an eine Benutzerkennung gekoppelt, siehe Abschnitt „Abläufe bei der Zugangsprüfung“ auf Seite 46. Daher ist es im Normalfall die Aufgabe jedes Benutzers, seine eigenen Berechtigungsprofile zu verwalten (anlegen, anzeigen, ändern und löschen).

Ein Berechtigungsprofil enthält unter anderem:

- eine Zugangsberechtigung. Diese Zugangsberechtigung muss eindeutig sein. Wenn ein Auftrag mit dem Berechtigungsprofil arbeiten soll, muss diese Zugangsberechtigung angegeben werden. FTAC erlaubt für diesen Auftrag dann nur die Zugriffsrechte, die im Berechtigungsprofil definiert sind. Um die Verantwortung für Aufträge eindeutig zuordnen zu können, empfiehlt es sich, eine Zugangsberechtigung für genau eine Person in genau einem Partnersystem vorzusehen.
- gegebenenfalls Vorgaben für Übertragungsaufträge wie z.B. Dateinamen, Dateinamen-Präfix, Folgeverarbeitungskommandos oder Präfix und/oder Suffix für Folgeverarbeitungskommandos.
- gegebenenfalls Angaben, welche Partnersysteme auf dieses Berechtigungsprofil zugreifen dürfen.
- Vorgaben zu erlaubten FT-Funktionen und zu Übertragungsrichtung, Schreibregel oder Verschlüsselung.
- ggf. Angaben, ob oder bis wann das Berechtigungsprofil genutzt werden kann.
- Angaben, ob und in wie weit das Profil Vorgaben des Berechtigungssatzes ignorieren kann. Der Benutzer darf jederzeit seine eigenen Vorgaben ignorieren. Wenn die Vorgaben des FTAC-Administrators ignoriert werden sollen, muss es privilegiert werden, siehe „Privilegierte Berechtigungsprofile“ auf Seite 122.

*Beispiel für Dateinamen-Präfix*

Ein Dateinamen-Präfix enthält einen Teil eines Datei- oder Pfadnamens. Der Benutzer des Profils kann sich dann nur unterhalb des angegebenen Datei- oder Pfadnamens bewegen.

- Auf einem Windows-System bedeutet C:\Users\Hugo\ als Dateinamen-Präfix, dass der Benutzer dieses Profils nur auf Verzeichnisse und Dateien unterhalb von C:\Users\Hugo\ zugreifen darf. Analoges gilt auf einem Unix-System, wenn z.B. /home/hugo/ als Dateinamen-Präfix angegeben wird.
- Wenn man im BS2000 beispielsweise PREFIX = USER. angibt, greift ein FT-Auftrag, in dem FILE-NAME = HUGO angegeben wurde, auf die Datei USER.HUGO zu.
- Auf z/OS z.B. versteht man unter einem Dateinamen-Präfix den "First-level Qualifier" und gegebenenfalls noch einen oder mehrere weitere Qualifier, z.B. 'OPUSERS.HUGO.NEU.'.

Auf diese Weise wird ausgeschlossen, dass sich jemand in gesperrte Verzeichnisse bewegen kann, oder dass jemand mit diesem Profil die Vorverarbeitungsfunktion nutzen kann. Als Dateinamen-Präfix kann aber auch ein fernes Vorverarbeitungskommando angegeben werden; im Auftrag sind dann z.B. nur noch Parameter zu diesem Kommando anzugeben.

### Auswirkungen eines Berechtigungsprofils

Die folgende Tabelle enthält in der linken Spalte die möglichen Einschränkungen der Zugriffsrechte in einem Berechtigungsprofil und in der rechten Spalte die Angaben, die für den Übertragungsauftrag in Bezug auf das Partnersystem nötig sind. Für ein Standard-Berechtigungsprofil gibt es einige Unterschiede, siehe oben.

| **Festlegung im Berechtigungsprofil** | **Angaben für den Übertragungsauftrag** |
|---|---|
| Zugangsberechtigung | Die Zugangsberechtigung adressiert das Berechtigungsprofil. Mit der Angabe von Benutzerkennung und Passwort kann nur das Standard-Berechtigungsprofil des Benutzers angesprochen werden, sofern dies definiert ist. |
| Übertragungsrichtung eingeschränkt | Die Angabe muss spiegelbildlich zur Festlegung im Berechtigungsprofil erfolgen. Wenn im Profil die Übertragungsrichtung "Vom Partner" steht, darf das ferne System nur Daten zum lokalen Rechner schicken, bei "An Partner" dürfen nur Dateien zum fernen System übertragen werden, im lokalen Rechner sind somit nur lesende Zugriffe erlaubt. |
| Partnersysteme vorgegeben | Der Auftrag kann nur von den Partnersystemen gestellt werden, die im Profil eingetragen sind. |
| Dateiname vorgegeben | Im Auftrag muss der Dateiname weggelassen werden.<br>Falls er im File-Transfer-Produkt des Partnersystems Pflichtparameter ist, muss er mit "*not-specified" belegt werden (z.B. BS2000-Systeme). |
| Präfix für den Dateinamen vorgegeben | Im Auftrag steht nur ein Teil des Dateinamens. FTAC ergänzt diese Angabe um das im Profil definierte Präfix zum vollständigen Dateinamen. Die Angabe absoluter Dateinamen oder das Verlassen des Verzeichnisses mittels ".." wird von FTAC unterbunden. |
| Verarbeitung nicht erlaubt | Im Auftrag darf für Ihren Rechner keine Verarbeitung verlangt werden. |
| Verarbeitung vorgegeben | Im Auftrag darf für Ihren Rechner keine Verarbeitung verlangt werden. |
| Präfix/Suffix für die Folgeverarbeitung vorgegeben | Im Auftrag darf nur der Teil der Folgeverarbeitung angegeben werden, der nicht im Profil steht. FTAC ergänzt diese Angabe zum vollständigen Folgeverarbeitungskommando. Wenn im Auftrag keine Folgeverarbeitung angegeben wurde, wird auch keine durchgeführt. |

| Festlegung im Berechtigungsprofil | Angaben für den Übertragungsauftrag |
|---|---|
| Einschränkung der Schreibregel | Der Auftrag wird nur dann durchgeführt, wenn er nicht gegen diese Schreibregel verstößt. |
| Verschlüsselung erzwingen oder verbieten | Der Auftrag wird nur dann durchgeführt, wenn er der Vorgabe des Berechtigungsprofils entspricht. |

### Privilegierte Berechtigungsprofile

Ein Berechtigungsprofil wird als privilegiert bezeichnet, wenn sich der Eigentümer des Profils über die (Administrator-)Vorgaben seines Berechtigungssatzes hinwegsetzen darf. Ein privilegiertes Berechtigungsprofil ist nur für Ausnahmefälle vorgesehen, z.B.:

– es soll eine bestimmte Datei übertragen werden,
– es ist keine oder nur eine bestimmte Folgeverarbeitung erlaubt,
– ein Partnersystem mit einer hohen Sicherheitsstufe darf mit der Benutzerkennung File-Transfer betreiben, andere mit kleineren Sicherheitsstufen aber nicht.

In einem privilegierten Berechtigungsprofil dürfen vom Benutzer nur die Zugangsberechtigung geändert und die Privilegierung wieder zurückgesetzt werden. Damit wird ein Missbrauch eines einmal privilegierten Berechtigungsprofils ausgeschlossen.

Ein Berechtigungsprofil kann nur durch den FTAC-Verwalter privilegiert werden.

### Hinweise zum Standard-Berechtigungsprofil

Ein Standard-Berechtigungsprofil besitzt im Gegensatz zu einem normalen Profil keine FTAC-Zugangsberechtigung, da der Zugang implizit über Benutzerkennung und Passwort geregelt ist. Andererseits können über dieses Profil die meisten der üblichen Parameter eingestellt werden wie z.B. die erlaubte FT-Funktion, ein Dateinamen-Präfix oder die Schreibregel. Nicht einstellbar sind die Nutzungsfrist, ob das Profil gesperrt ist oder nicht und ob das Profil privat oder öffentlich ist.

Ein Standard-Berechtigungsprofil muss explizit eingerichtet werden, pro Benutzerkennung ist höchstens ein Standard-Berechtigungsprofil möglich.

### 5.1.3 Die FTAC-Logging-Funktion

Jeden FT-Auftrag, an dem das geschützte System beteiligt ist, unterzieht openFT-AC einer Zugangsprüfung und protokolliert deren Ergebnis. Die Informationen werden in den sogenannten FTAC-Logging-Sätzen abgelegt. FTAC-Logging-Sätze enthalten neben den üblichen Logging-Daten (Zeitstempel etc.) folgende, für den FTAC-Verwalter wichtige Information:

- Funktion des jeweiligen FT-Auftrags
- Grund für eine eventuelle Zurückweisung des Auftrages durch FTAC.
- Übertragungsrichtung des FT-Auftrags
- Name des Partnersystems, mit dem der FT-Auftrag durchgeführt wurde bzw. werden sollte
- LOGON-Berechtigung (USER-IDENTIFICATION) des Auftraggebers bei Aufträgen, die im lokalen System gestellt wurden (ansonsten *REMOTE für ferne Auftraggeber)
- Name und Privilegierungskennzeichen eines ggf. benutzten Berechtigungsprofils
- den lokalen Dateinamen oder Bibliotheksnamen (auf BS2000 und z/OS)

FTAC prüft lediglich die Berechtigung eines Auftrages anhand der Berechtigungssätze und -profile. Ob dieser Auftrag dann auch tatsächlich von openFT ausgeführt werden kann, protokolliert openFT in den FT- oder ADM-Logging-Sätzen.

Die Ausgabe von FTAC-Logging-Sätzen kann nicht ausgeschaltet werden. Sie kann aber eingeschränkt werden auf von FTAC abgelehnte Aufträge oder auf modifizierende Aufträge.

Der FTAC-Verwalter kann sich über alle Zugangsprüfungen informieren, die FTAC bislang durchgeführt hat. Dadurch wird beispielsweise eine Revision des Systems erleichtert.

#### Löschen von Logging-Sätzen

FT- und FTAC-Verwalter sind die einzigen Benutzer im System, die sich alle FTAC-Logging-Sätze ansehen können und sie auch löschen dürfen. Der FT-Benutzer kann nur seine eigenen Logging-Sätze ansehen, er darf keine Logging-Sätze löschen.

Es können nur FTAC-Logging-Sätze vom ältesten Datum bis zu einem ausgewählten Datum gelöscht werden. Damit liegen die FTAC-Logging-Sätze immer lückenlos bis zum aktuellsten Satz in der Protokolldatei vor.

Im Prinzip schreibt FTAC beliebig viele Logging-Sätze („bis die Platte voll ist“). Von Zeit zu Zeit sollten Sie als FTAC-Verwalter die vorhandenen Logging-Sätze sichern (z.B. als Ausdruck, auf Band oder als Datei im CSV-Format) und anschließend diese Logging-Sätze aus der aktuellen Logging-Datei entfernen. Dadurch bleiben zum einen die Logging-Sätze für eine lückenlose Dokumentation über einen längeren Zeitraum erhalten, und zum anderen wird die Logging-Datei nicht überflüssig groß. Der FTAC-Verwalter kann die aktuelle Logging-Datei wechseln und ältere Logging-Sätze in Offline-Logging-Dateien vorhalten.

## 5.2 Authentifizierung

Das in diesem Abschnitt beschrieben Konzept steht für openFT-Partner ab V8.1 für Unix-Systeme und Windows-Systeme und openFT ab V9.0 für BS2000 und z/OS zur Verfügung. Für FTAM-Partner und FTP-Partner steht die Authentifizierung in dieser Form nicht zur Verfügung, da weder das FTP-Protokoll noch das von der ISO genormte FTAM-Protokoll eine vergleichbare Funktionalität vorsehen.

Sollen sicherheitskritische Daten übertragen werden, so ist es wichtig, das jeweilige Partnersystem vor der Übertragung einer sicheren Identitätsprüfung zu unterziehen („Authentifizierung“). Die beiden an einer Übertragung beteiligten openFT-Instanzen müssen gegenseitig mit kryptografischen Mitteln überprüfen können, ob sie mit der „richtigen“ Partnerinstanz verbunden sind.

Das openFT-Konzept basiert auf der Adressierung der openFT-Instanzen durch netzweit eindeutige Identifikationen sowie dem Austausch von partnerspezifischen Schlüsselinformationen.

Bitte beachten Sie, dass die Authentifizierung nur für benannte Partner möglich ist!

### 5.2.1 Einsatzfälle für die Authentifizierung

Bei der Authentifizierung sind drei Einsatzfälle zu unterscheiden:

- Fall 1:
  Die lokale openFT-Instanz überprüft die Identität der Partnerinstanz. Das setzt voraus, dass lokal ein aktueller öffentlicher Schlüssel der Partnerinstanz abgelegt wurde, siehe Abschnitt „Schlüssel von Partnersystemen“ auf Seite 128.
  Eine derartige Konfiguration macht beispielsweise Sinn, wenn per openFT auf Dateien eines File Servers zugegriffen werden soll. Für die lokale openFT-Instanz ist es wichtig, dass die bezogenen Daten aus einer sicheren Quelle (dem authentifizierten Partner) kommen, umgekehrt ist es für den File Server unerheblich, wer dort zugreift.
- Fall 2:
  Die Partnerinstanz überprüft die Identität der lokalen openFT-Instanz. Das setzt voraus, dass ein aktueller öffentlicher Schlüssel der lokalen openFT-Instanz in der Partnerinstanz hinterlegt ist (bei Unix- und Windows-Partnern umcodiert), siehe Abschnitte „Lokale RSA-Schlüsselpaare“ auf Seite 126 und „Sicheres Verteilen von Schlüsseln“ auf Seite 129.
  Eine solche Konfiguration wäre beispielsweise denkbar, wenn von einem zentralen Rechner per openFT auf Partnersysteme in mehreren Filialen zugegriffen werden soll und die Filialrechner nur Zugriffe des Zentralrechners (und wirklich nur dieses Rechners) zulassen dürfen.

- Fall 3:
  Beide an einer Übertragung beteiligten openFT-Instanzen authentifizieren sich gegenseitig (Kombination aus Fall 1 und Fall 2). Das setzt voraus, dass gegenseitig aktuelle öffentliche Schlüssel ausgetauscht wurden und sich die Partner über ihre Instanzidentifikationen adressieren. Damit ist gewährleistet, dass die Daten sowohl aus einer sicheren Quelle stammen als auch nur in sichere Hände gelangen.

Bei Konfigurationsfehlern, die die Authentifizierung eines der an einem Auftrag beteiligten Partner verhindern, kommt keine Session zustande, in der der Auftrag durchgeführt werden kann. Daher lässt sich das Problem nicht am Auftragszustand, sondern am Partnerzustand ablesen. Der Partnerzustand (RAUTH oder LAUTH) gibt Auskunft darüber, auf welcher Seite das Problem erkannt wurde.

## 5.2.2 Instanzidentifikationen

Die Instanzidentifikation ist ein bis zu 64 Zeichen langer Name, dessen Eindeutigkeit unabhängig von Groß- und Kleinschreibung **netzweit** gelten muss. Sie spielt insbesondere dann eine Rolle, wenn mit Authentifizierung gearbeitet wird.

Bei der Installation wird standardmäßig der Name des lokalen Rechners im lokalen Netz als Instanzidentifikation festgelegt. Falls die netzweite Eindeutigkeit nicht gesichert ist, muss der FT-Verwalter die lokale Instanzidentifikation per Betriebsparametereinstellung ändern.

### Instanzidentifikation von Partnern

Instanzidentifikationen von Partnersystemen hinterlegen Sie in den Eigenschaften des Partnersystems. Daher muss das Partnersystem in der Partnerliste eingetragen sein, siehe Abschnitt „Partnerliste“ auf Seite 51. Anhand der Instanzidentifikationen der Partnersysteme verwaltet openFT die diesen Partnern zugeordneten Betriebsmittel wie z.B. Auftragswarteschlangen und kryptografische Schlüssel.

## 5.2.3 Lokale RSA-Schlüsselpaare

RSA-Schlüssel werden für die Authentifizierung und die Aushandlung des AES-Schlüssels verwendet, mit dem die Auftragsbeschreibungsdaten und Dateiinhalte verschlüsselt werden.

### 5.2.3.1 Eigenschaften von Schlüsselpaaren

Ein RSA-Schlüsselpaar besteht jeweils aus einem privaten (private key) und einem öffentlichen Schlüssel (public key). Es gibt bis zu drei Schlüsselpaarsätze bestehend aus jeweils drei Schlüsselpaaren in den Längen 768, 1024, 2048. Beim Erzeugen eines Schlüsselpaarsatzes werden immer neue Schlüsselpaare für jede dieser Längen erzeugt.

Die öffentlichen Schlüssel werden unter folgenden Namen hinterlegt:

SYSPKF.R<Schlüsselreferenz>.L<Schlüssellänge>.

Der Ablageort ist plattformabhängig, siehe jeweiliges openFT-Handbuch „Installation und Betrieb“.

Die Schlüsselreferenz ist ein numerischer Bezeichner für die Version des Schlüsselpaares. Die öffentlichen Schlüsseldateien sind Textdateien, die im Zeichencode des jeweiligen Betriebssystems erzeugt werden, d.h. standardmäßig:

- BS2000-Systeme: Wert der Systemvariable HOSTCODE
- z/OS: IBM1047
- Unix-Systeme: ISO8859-1
- Windows-Systeme: CP1252

Private Schlüssel werden von openFT intern verwaltet.

Für die Verschlüsselung wird standardmäßig ein Schlüssel der Länge 2048 verwendet. Diese Einstellung kann der FT-Verwalter per Betriebsparameter ändern.

#### Kommentare

Zu jedem Schlüsselpaarsatz können Kommentare hinterlegt werden, die beim Erzeugen eines Schlüsselpaarsatzes in die ersten Zeilen der öffentlichen Schlüsseldateien geschrieben werden. Kommentare könnten beispielsweise die Kontaktdaten des zuständigen FT-Verwalters, den Rechnernamen oder ähnliche für Partner wichtige Informationen enthalten. Die Kommentare in der editierbaren Textdatei SYSPKF.COMMENT dürfen maximal 78 Zeichen lang sein. Beim Aktualisieren eines Schlüsselpaarsatzes werden nachträglich aktualisierte Kommentare aus dieser Datei in existierende öffentliche Schlüsseldateien eingebracht.

#### 5.2.3.2 Schlüsseldateien aktualisieren und Schlüssel ersetzen

Die Verwendung von Schlüsseln bietet zusätzliche Sicherheit. Diese Sicherheit wird durch folgende Möglichkeiten auch auf Dauer gewährleistet:

- Die öffentlichen Schlüsseldateien der bestehenden Schlüsselpaarsätze können neu erzeugt werden. Dies ist z.B. notwendig, wenn eine öffentliche Schlüsseldatei versehentlich gelöscht oder anderweitig manipuliert wurde.
- Ein Schlüsselpaarsatz kann durch einen komplett neuen Schlüsselpaarsatz ersetzt werden. Der aktuellste öffentliche Schlüssel ist an der höchstwertigen Schlüsselreferenz im Namen der Datei zu erkennen. openFT unterstützt maximal drei Schlüsselpaarsätze gleichzeitig. Mehrere Schlüssel sollten aber nur so lange existieren, bis allen Partnersystemen der aktuellste öffentliche Schlüssel zur Verfügung gestellt wurde.
- Nicht mehr benötigte Schlüsselpaarsätze können (und sollten!) umgehend gelöscht werden.

#### 5.2.3.3 Importierte Schlüssel

Schlüssel bestehen aus Textdateien. Im Prinzip ist es daher möglich, diese Dateien mit Betriebssystemmitteln („händisch“) in das lokale System einzubringen und an die Stelle zu kopieren, an der openFT die Schlüssel erwartet. Diese Methode ist umständlich und fehleranfällig, außerdem würden auf einigen Systemen besondere Verwalterrechte benötigt.

Daher bietet die openFT eigene Funktionen an, mit denen sich folgende Schlüssel importieren lassen:

- Öffentliche Schlüssel von Partnerinstanzen. Diese Schlüssel müssen von der openFT-Instanz des Partners erzeugt worden sein.
- Private Schlüssel, die mit einem externen Tool (d.h. nicht über openFT) erzeugt wurden. openFT erzeugt beim Importieren eines privaten Schlüssels den zugehörigen öffentlichen Schlüssel. Dieser Schlüssel kann wie ein mit openFT erzeugter Schlüssel verwendet und an Partnersysteme verteilt werden.

Die Import-Funktionen haben gegenüber der händischen Methode den Vorteil, dass die Schlüssel (auch die neu erzeugten) gleich an der richtigen Stelle im lokalen System abgelegt werden.

#### Schlüsselformate

openFT unterstützt Schlüsseldateien in den folgenden Formaten:

- PEM-Format (native PEM)

  Die PEM-codierten Dateien müssen im EBCDIC-Format vorliegen.

- PKCS#8 Format ohne Passphrase oder nach v1/v2 mit einer Passphrase verschlüsselt (PEM-codiert).

- PKCS#12 v1 Format in Form einer Binärdatei. Die Datei wird nach einem privaten Schlüssel durchsucht, nicht unterstützte Bestandteile (z.B. Zertifikate, CRLs) werden beim Import ignoriert. Ist das Zertifikat per Signatur oder Hash geschützt, so wird von openFT keine Gültigkeitsprüfung durchgeführt. Die Gültigkeit der Datei muss durch externe Mittel sichergestellt werden. Der erste private Schlüssel, der in der Datei gefunden wird, wird importiert, weitere werden ignoriert.

Eine zur Verschlüsselung verwendete Passphrase muss beim Importieren im Passwort-Parameter angegeben werden.

### 5.2.4 Schlüssel von Partnersystemen

Schlüssel von Partnersystemen können nur dann importiert werden, wenn das Partnersystem als benannter Partner (siehe Seite 52) in der Partnerliste eingetragen ist, denn der Partnername wird als Referenz verwendet.

Beim Importieren sollte möglichst die Import-Funktion von openFT verwendet werden.

#### Eigenschaften von Partnersystem-Schlüsseln

Sie können für die Schlüssel von Partnersystemen folgende Eigenschaften zuordnen:

- ein Verfallsdatum, d.h. der Schlüssel kann nach Ablauf dieses Datums nicht mehr verwendet werden.

- Authentifizierungsstufe (1 oder 2): Mit Authentifizierungsstufe 2 führt openFT zusätzliche interne Prüfungen durch. Stufe 2 wird für alle openFT-Partner ab Version 11.0B unterstützt. Ein Authentifizierungsversuch nach Stufe 1 wird zu diesem Partner abgelehnt.

Diese Eigenschaften lassen sich für einen bestimmten Partner oder für alle Partner festlegen.

### 5.2.5 Sicheres Verteilen von Schlüsseln

Die Verteilung der öffentlichen Schlüsseldateien zwischen den Partnersystemen sollte auf gesichertem Weg geschehen, also z.B. durch

– kryptografisch abgesicherte Verteilung per E-Mail
– Verteilung per CD (persönliche Übergabe oder per Einschreiben)
– Verteilung über einen zentralen openFT-Fileserver, dessen öffentlichen Schlüssel die Partner besitzen.

Wenn Sie öffentliche Schlüsseldateien zwischen Partnersystemen mit unterschiedlichem Betriebssystem verteilen, müssen Sie darauf achten, dass diese Dateien umcodiert werden, z.B. von EBCDIC.DF04-1 (BS2000-System) nach ISO 8859-1 (Unix-System) oder CP1252 (Windows-System). Wenn Sie die Schlüsseldatei mit openFT als Textdatei übertragen, dann wird sie automatisch korrekt umcodiert.

## 5.3 Erweiterte Absenderüberprüfung

Die erweiterte Absenderüberprüfung gilt für Partnersysteme, die nicht mit Authentifizierung arbeiten. Bei inbound-Aufträgen wird anhand der Instanzidentifikation überprüft, ob das „rufende“ System über einen gültigen Eintrag in der Partnerliste verfügt. openFT bietet mit der erweiterten Absenderüberprüfung die Möglichkeit, nicht nur die Instanzidentifikation, sondern zusätzlich auch die Transportadresse zu überprüfen.

Die erweiterte Absenderüberprüfung kann für openFT-Partner global per Betriebsparameter oder partnerspezifisch eingeschaltet werden.

Die globale Einstellung gilt für alle Partner, für die der Automatismus „Partnerattribute“ eingeschaltet ist.

Für dynamische Partner hat die erweiterte Absenderüberprüfung keine Bedeutung, da diese immer über die Transportadresse identifiziert werden.

Fällt die Absenderüberprüfung negativ aus, wird der Auftrag abgewiesen.

### Besonderheiten bei FTAM-Partnern

Bei BS2000-Systemen läuft die Absenderüberprüfung bei FTAM-Partnern stets über die Transportadresse. Auf Windows- und Unix-Systemen kann ein FTAM-Partner zusätzlich durch den von der Partneradresse unabhängigen Application Entity Title (AET) identifiziert werden. Dazu kann in den Betriebsparametern eingestellt werden, ob der Absender durch die Transportadresse, durch den AET oder durch beides geprüft wird. Eine partnerspezifische Einstellung ist für FTAM-Partner nicht möglich.

### Besonderheiten bei FTP-Partnern

Bei FTP-Partnern läuft die Absenderüberprüfung ausschließlich über die Transportadresse. Deshalb bleibt die Eigenschaft "erweiterte Absenderüberprüfung" für FTP-Partner wirkungslos und wird auch nicht angezeigt.

## 5.4 Verschlüsselung bei der Datenübertragung

Bei der Dateiübertragung verschlüsselt openFT standardmäßig die Auftragsbeschreibungsdaten. Optional können auch Dateiinhalte verschlüsselt werden.

### 5.4.1 Verschlüsselung der Auftragsbeschreibungsdaten

Die beteiligten Partner handeln die Verschlüsselung, den Verschlüsselungsalgorithmus und den verwendeten Schlüssel beim Verbindungsaufbau aus.

openFT verwendet für die Verschlüsselung nach Möglichkeit das Verfahren RSA/AES mit einer AES-Schlüssellänge von 256 Bit. Bei der Kopplung zu älteren Partnern kann auch RSA/AES mit 128 Bit bzw. RSA/DES zum Einsatz kommen. Es wird jeweils das sicherste, von beiden Partnern unterstützte Verfahren verwendet.

Zusätzlich per Betriebsparameter kann ein minimaler AES-Schlüssel vorgegeben werden, d.h. es werden nur AES-Schlüssel der vorgegebenen Länge oder längere akzeptiert. Kann der Partner diese Anforderung nicht erfüllen, dann wird der Auftrag abgelehnt.

openFT verschlüsselt automatisch die Auftragsbeschreibungsdaten, sofern beide Partner diese Funktionalität unterstützen, im lokalen System ein RSA-Schlüsselpaarsatz existiert und die Verschlüsselung nicht explizit abgeschaltet wurde. Sie können als FT-Verwalter die gewünschte Schlüssellänge des RSA-Schlüssels (RSA-PROPOSED) per Betriebsparameter einstellen. Der Standardwert nach Installation ist 2048. Außerdem können Sie per Betriebsparameter eine Mindestschlüssellänge einstellen.

Hat eine der beteiligten Instanzen eine RSA-Mindestschlüssellänge konfiguriert, so ist sichergestellt, dass die Aushandlung des AES-Schlüssels mit dieser Mindestschlüssellänge ausgeführt wird.

Hat einer der Partner entweder keinen gültigen RSA-Schlüssel, oder ist die Verschlüsselung ausgeschaltet, während der andere Partner eine Mindestschlüssellänge verlangt, wird keine Verbindung zwischen den beiden Partnern möglich sein.

### 5.4.2 Verschlüsselung der Dateiinhalte

Die Verschlüsselung der Dateiinhalte (d.h. Benutzerdaten) ist nur möglich, wenn openFT-CR installiert wurde. Dieses Produkt unterliegt Exportbeschränkungen und ist daher nicht in allen Staaten verfügbar. In BS2000-Systemen wird das Produkt CRYPT zum Verschlüsseln verwendet, sofern es installiert und gestartet ist. Andernfalls werden die openFT-internen Verschlüsselungsalgorithmen verwendet.

Die Verschlüsselung von Benutzerdaten steht nur zur Verfügung für:

- Übertragungsaufträge mit openFT-Partnern
- Outbound-Aufträge über das TLS-Protokoll an einen FTP-Server, der Secure FTP unterstützt

Ist eines der beiden Systeme (lokales System oder Partnersystem) nicht zur verschlüsselten Dateiübertragung bereit, wird der Auftrag abgelehnt.

### Einstellmöglichkeiten für Dateiinhalt-Verschlüsselung

Mit openFT können Sie

- bei Outbound-Aufträgen gezielt eine verschlüsselte Übertragung Ihrer Benutzerdaten anfordern
- bei Inbound-Aufträgen die Verschlüsselung der Benutzerdaten über ein Berechtigungsprofil erzwingen oder verbieten:
  - Die Verschlüsselung kann explizit erzwungen werden, z.B. für besonders sicherheitsrelevante Aufträge. Aufträge ohne Verschlüsselung der Benutzerdaten werden abgelehnt.
  - Die Verschlüsselung kann explizit verboten werden, z.B. für weniger sicherheitsrelevante Aufträge, bei denen es auf Performance ankommt. Aufträge mit Verschlüsselung der Benutzerdaten werden abgelehnt.
- als FT-Verwalter per Betriebsparameter-Einstellung die Datenverschlüsselung für Inbound- und/oder Outbound-Aufträge generell erzwingen oder einen minimale Länge des AES-Schlüssels vorgeben. Ist eine Mindestschlüssellänge für den RSA- und/oder AES-Schlüssel eingestellt, dann ist das Verhalten dasselbe wie wie im Abschnitt „Verschlüsselung der Auftragsbeschreibungsdaten“ auf Seite 131 beschrieben.

  Die Einstellungen gelten für Dateiübertragungsaufträge über das openFT-Protokoll und für Administrationsaufträge. FTAM-Aufträge und inbound FTP-Aufträge werden abgelehnt, da keine Verschlüsselung unterstützt wird. Dateimanagement wird unabhängig von den Einstellungen unverschlüsselt durchgeführt. Zusätzlich gilt:
  - Ist die Outbound-Verschlüsselung aktiviert, dann wird bei einem Outbound-Auftrag der Dateiinhalt verschlüsselt, auch wenn im Auftrag selber keine Verschlüsselung angefordert wurde. Wenn der Partner keine Verschlüsselung unterstützt (z.B. weil sie ausgeschaltet ist oder openFT-CR nicht installiert ist), dann wird der Auftrag abgelehnt.
  - Ist die Inbound-Verschlüsselung aktiviert und soll ein unverschlüsselter Inbound-Auftrag bearbeitet werden, dann wird dieser Auftrag abgelehnt.

Beachten Sie bitte, dass der Aufwand für die Verschlüsselung in den beteiligten Partnersystemen Performance kostet.

## 5.5 Schutzmechanismen gegen Datenmanipulation

openFT prüft bei der Kommunikation mit openFT-Partnern ab V8.1 implizit die Integrität der übertragenen Daten. Die Integrität der Auftragsbeschreibungsdaten wird immer überprüft . Abhängig von Optionen, die im Auftrag oder per Betriebsparameter eingestellt sind, gilt Folgendes:

- Bei Aufträgen mit Verschlüsselung wird auch die Integrität des übertragenen Dateiinhalts geprüft.
- Bei Aufträgen ohne Verschlüsselung kann im Auftrag explizit die Überprüfung des Dateiinhalts angefordert werden.

Wird ein Fehler erkannt, versuchen wiederanlauffähige Aufträge eine erneute Übertragung. Nicht wiederanlauffähige Aufträge werden abgebrochen.

Auf diese Weise kann eine böswillige Manipulation (z.B. in unsicheren offenen Netzen wie dem Internet) der übertragenen Daten erkannt und verhindert werden.

Fehler auf den physikalischen Übertragungswegen werden vom Kommunikationssystem selbst erkannt und behoben. Hierfür ist auf openFT Ebene keine Datenintegritätsprüfung notwendig.

## 5.6 Verschlüsselung mit FTPS

Um eine FTP-Übertragung abzusichern, kann die Transportverbindung durch Transport Layer Security (TLS) verschlüsselt werden. Dieses Protokoll ist im RFC4217 (Securing FTP with TLS) beschrieben und wird meist als FTPS bezeichnet. Auch die Bezeichnungen *FTP Secure*, *FTP über TLS* oder *FTP über SSL* (Name des Vorgängerprotokolls) sind gebräuchlich.

Für die Verschlüsselung stellt ein FTPS-Server dem openFT seinen Schlüssel und sein Zertifikat zur Verfügung. Eine gegenseitige Authentifizierung findet nicht statt.

openFT kann als Client outbound verschlüsselte Benutzerdaten mit einem FTPS-Server austauschen, wenn auf openFT-Seite openFT-CR installiert ist und der FTP-Server das Protokoll TLS unterstützt. Als Verschlüsselungsverfahren wird AES verwendet. Wenn der openFT-Client im Auftrag die Verschlüsselung der Benutzerdaten verlangt, der FTP-Server das Protokoll TLS aber nicht unterstützt, dann wird der Auftrag abgelehnt.

Wenn der openFT-Client keine Verschlüsselung der Benutzerdaten verlangt, dann werden die Auftragsbeschreibungsdaten verschlüsselt, wenn der FTP-Server das Protokoll TLS akzeptiert, sonst werden die Auftragsbeschreibungsdaten unverschlüsselt übertragen.

# 6 Bedienung von openFT

openFT bietet für den Anwender folgende Schnittstellen:

- Kommandoschnittstelle
- openFT Explorer für Unix- und Windows-Systeme
- Programmschnittstelle
- openFT-Script-Schnittstelle
- ISPF-Panels für z/OS

## 6.1 Kommandoschnittstelle

openFT bietet auf allen Plattformen eine Kommandoschnittstelle. Damit lassen sich die openFT-Funktionen über Prozeduren oder Skripte erledigen, d.h. der Betrieb wird automatisiert und die Aufgaben werden im Batch-Betrieb zu bestimmten Zeiten oder bei bestimmten Ereignissen ausgeführt.

Die Kommandoschnittstelle bietet Returncodes, so dass auf Fehler automatisch reagiert werden kann, z.B. wenn ein Partnersystem nicht erreichbar ist.

Die Kommandoschnittstelle gibt es in zwei Ausprägungen:

- Auf den BS2000-Systemen stehen die Kommandos im SDF-Format zur Verfügung, d.h. dem Standardformat im BS2000.

  Auf z/OS-Systemen stehen die Kommandos ebenfalls in SDF-Syntax zur Verfügung. Die Syntax ist auf BS2000- und z/OS-Systemen bis auf die Kommandonamen identisch: Auf z/OS-Systemen sind systembedingt nur kurze Namen möglich (max. 8 Zeichen), während auf BS2000-Systemen die Namen in der Regel sehr ausführlich sind.

  Bei den Funktionen gibt es nur kleine plattformspezifische Unterschiede.

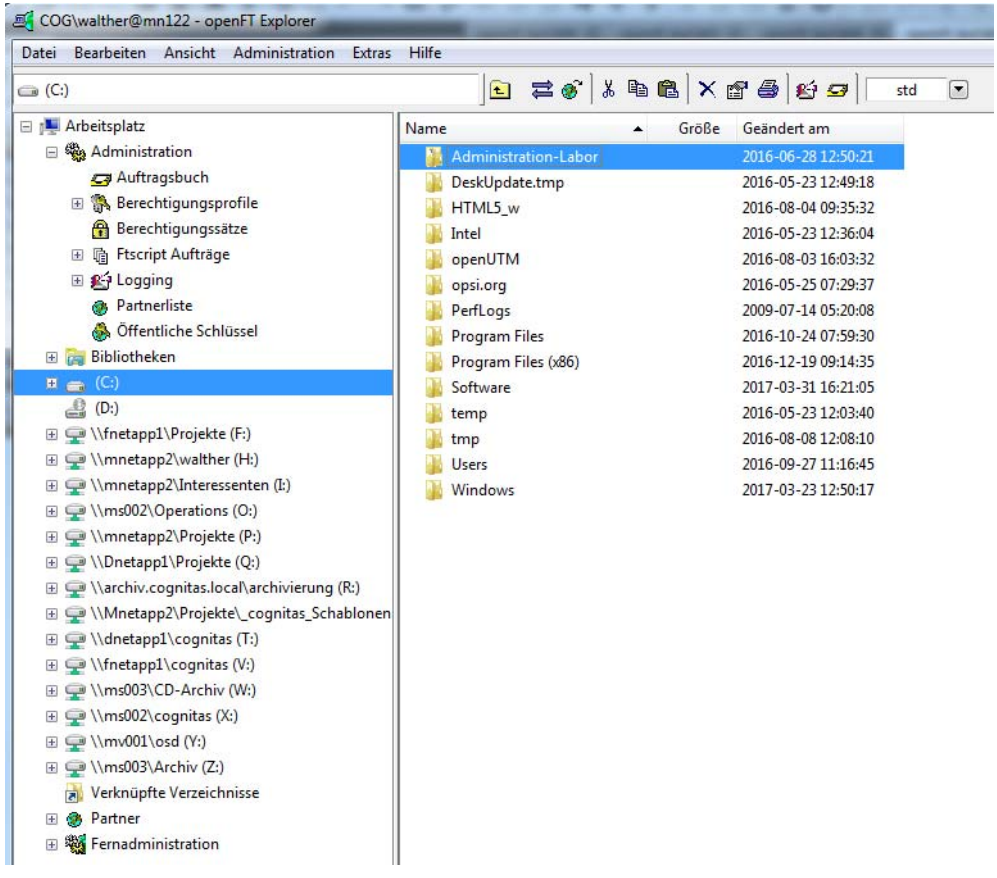
- Auf Unix- und Windows-Systemen stehen die Kommandos in dem für Unix- und Windows-Systeme üblichen Format zur Verfügung. Die Syntax ist auf beiden Plattformen identisch, bei den Funktionen gibt es nur geringe Unterschiede. Die Syntax wird auch an der BS2000-POSIX-Schnittstelle unterstützt.

  Auf Unix-Systemen ist zu jedem Kommando ist auch eine Manpage verfügbar, die Sie von der Kommandozeile aus aufrufen können. Verwenden Sie dazu das Unix-Kommando *man* zusammen mit dem Namen eines Kommandos als Argument (z.B. *man ft*).

Weitere Details siehe Handbuch "openFT (BS2000) - Kommandoschnittstelle", Handbuch "openFT (z/OS) - Kommandoschnittstelle" und Handbuch "openFT (Unix- und Windows-Systeme) - Kommandoschnittstelle".

## 6.2 openFT Explorer für Unix- und Windows-Systeme

Der openFT Explorer ist eine grafische Oberfläche, die im Aussehen und der Bedienung dem Windows Explorer nachempfunden ist. Im linken Teil des Fensters erscheinen die Objektverzeichnisse und im rechten Fenster die Objekte des markierten Objektverzeichnisses, siehe folgendes Beispiel für Windows-Systeme:




Mit dem openFT Explorer können alle Funktionen der Kommandoschnittstelle ausgeführt werden, d.h. Verzeichnisübertragung, Dateiübertragung, Dateimanagement, ferne Kommandoausführung sowie alle Administrationsaufgaben. Er ist besonders geeignet für Administrationsaufgaben, die sich nicht ständig wiederholen oder nicht so einfach automatisieren lassen.

Neben den openFT-Funktionen der Kommandoschnittstelle bietet der openFT Explorer eine Reihe von zusätzlichen Funktionen wie z.B.:

- Konfigurieren der Zentralen Administration. Dazu stellt der openFT-Explorer den integrierten Konfigurations-Editor zur Verfügung.

- Remote Administration
- Editieren von Textdateien mit dem integrierten Texteditor
- Dateibaumabgleich
- Einfache Datei- und Verzeichnisübertragung mit Copy und Paste oder Verschieben per Maus.
- Benutzerspezifisches Partnerverzeichnis: Ein Benutzer kann sich eigene Listen mit häufig verwendeten Partnern anlegen.

## 6.3 Programmschnittstelle

Die openFT-Funktionen können über eine plattform-spezifische Programmschnittstelle aufgerufen werden. Damit ist z.B. eine Integration von openFT-Funktionen in bestehende Programme möglich.

Die Programmschnittstelle gibt es in folgenden Ausprägungen:

- BS2000-Systeme: Assembler-Makros und Cobol-Aufrufe.

  Beide Schnittstellen umfassen die Dateiübertragung, das Dateimanagement und die komplette FTAC-Funktion. Weitere Details siehe Handbuch "openFT (BS2000) - Programmschnittstelle".

- z/OS: Assembler-Makro

  Der Makro verarbeitet den String eines openFT-Kommandos. Damit kann der komplette Funktionsumfang der Kommandoschnittstelle genutzt werden. Weitere Details siehe Handbuch "openFT (z/OS) - Kommandoschnittstelle".

- Unix- und Windows-Systeme: C und Java

  Beide Schnittstellen umfassen die Dateiübertragung, das Dateimanagement und das Ausführen ferner Kommandos (ftexec-Funktion). Weitere Details siehe Handbuch "openFT (Unix- und Windows-Systeme) - Programmschnittstelle".

- Windows-Systeme: OCX Control

  Diese Schnittstelle bietet die Möglichkeit, synchrone Datenübertragungsanforderungen über Standardschnittstellen (OLE/COM) von Anwendungsprogrammen auszuführen. Weitere Details siehe Handbuch "openFT (Unix- und Windows-Systeme) - Programmschnittstelle".

## 6.4 openFT-Script-Schnittstelle

openFT-Script stellt eine Skript-Sprache in XML-Notation zur Verfügung. Diese umfasst von der Kommando- oder C-Schnittstelle gewohnte openFT-Funktionen und bietet zusätzliche Kontextverwaltungs- und Steuerungsfunktionen.

Mit openFT-Script können mehrere logisch voneinander abhängige openFT-Aufträge zu einem Auftrag (Ftscript) zusammengefasst werden. openFT-Script entlastet den Anwender beim Überwachen zusammenhängender openFT-Aufträge und bietet einen Wiederanlauf im Fall einer Unterbrechung.

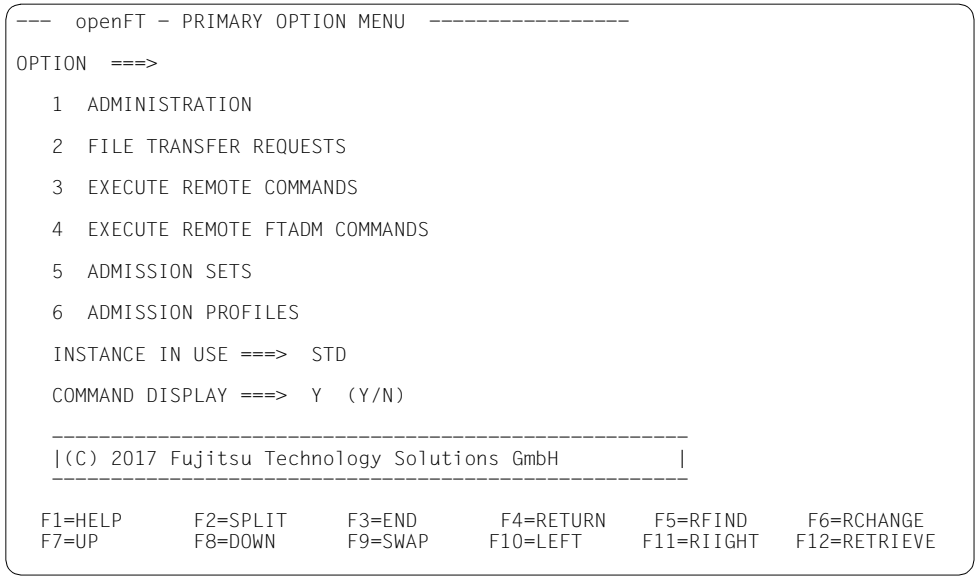
Die XML-Anweisungen eines openFT-Script-Auftrages werden in einer Textdatei hinterlegt. Diese Dateien können mit einem Texteditor oder beliebigen XML-Werkzeugen bearbeitet werden, ein Compiler ist nicht notwendig. Für den Ablauf muss auf Ihrem System mindestens das J2SE™ Runtime Environment 7.0 (JRE 7.0) installiert sein.

Gestartet wird ein openFT-Script-Auftrag per Kommando *ftscript*. Zusätzlich bietet die openFT-Script-Schnittstelle weitere Kommandos zum Verwalten von openFT-Script-Läufen. openFT-Script-Aufträge können auch im openFT Explorer über das Objektverzeichnis **Ftscript Aufträge** überwacht und abgebrochen werden.

Eine detaillierte Beschreibung der XML-Schnittstelle finden Sie im Handbuch "openFT-Script-Schnittstelle".

## 6.5 ISPF-Panels für z/OS

Auf z/OS kann openFT über so genannte ISPF-Panels bedient werden. Der Aufruf der openFT-Funktionen über ISPF-Panels setzt voraus, dass auf der Anlage das IBM Program Product "Interactive System Productivity Facility" (ISPF) installiert ist. ISPF stellt eine zeichenorientierte Oberfläche in Form von so genannten „Panels“ zur Verfügung. Die Darstellung entspricht der einer 3270-Terminal-Emulation, d.h. standardmäßig 24 Zeilen mit je 80 Zeichen. Das folgende Beispiel zeigt ein typisches „Startmenü“:




Die Panels bieten eine Menü- und Fragebogentechnik mit sofortigen Fehlerhinweisen und Hilfefunktionen etc. an. Mit Hilfe der Panels können nahezu alle openFT-Funktionen genutzt werden, die auch an der Kommandoschnittstelle zur Verfügung stehen.

Wenn der FT-Benutzer diese Menüschnittstelle verwendet, werden intern die entsprechenden Kommandos abgesetzt.

# 7 Administration

Die Administration von openFT umfasst folgende Themen:

- Rollenkonzept für die Administration
- Lokale Administration
- Remote Administration über openFT Explorer
- Zentrale Administration

## 7.1 Rollenkonzept für die Administration

Die Verwaltung von openFT wird auf mehrere Rollen aufgeteilt, so dass die einzelnen Aufgaben von unterschiedlichen Personen übernommen werden können.

openFT kennt folgende Rollen.

- FT-Verwalter

  Der FT-Verwalter überwacht und steuert den FT-Betrieb, d.h. Starten und Beenden von openFT, Betriebsparameter einstellen, Partner verwalten etc.

- FTAC-Verwalter

  Der FTAC-Verwalter überwacht den FTAC-Betrieb, d.h. Berechtigungssätze und Berechtigungsprofile verwalten und sichern.

- ADM-Verwalter

  Der ADM-Verwalter verwaltet den Fernadministrations-Server. Daher gibt es den ADM-Verwalter nur auf Unix- und Windows-Systemen, siehe auch Abschnitt „Fernadministration“ auf Seite 160.

### Wer ist FT-Verwalter und FTAC-Verwalter?

Die Rolle von FT- und FTAC-Verwalter sind auf allen Plattformen an besondere Rechte gebunden. Wie die beiden Rollen konkreten Benutzerkennungen zugeordnet werden, hängt von der Plattform ab:

- BS2000-Systeme:

  Der FT-Verwalter und der FTAC-Verwalter werden durch die BS2000-Privilegien FT-ADMINISTRATION und FTAC-ADMINISTRATION festgelegt. Beide Privilegien liegen standardmäßig bei der BS2000-Systemverwalterkennung TSOS. Sofern auf der BS2000-System SECOS im Einsatz ist, können beide Privilegien an eine andere Kennung weitergegeben werden.

  Weitere Details siehe Handbuch "openFT (BS2000) - Installation und Betrieb".

- z/OS-Systeme:

  Die Benutzerkennungen für FT- und FTAC-Verwalter werden in einer Konfigurationsdatei festgelegt, der so genannten FT-Parameterbibliothek. Diese Konfigurationsdatei darf nur der Systemverwalter editieren. Auf z/OS kann die Rolle des FT- und des FTAC-Verwalters jeweils an mehrere Kennungen vergeben werden.

  Weitere Details siehe Handbuch "openFT (z/OS) - Installation und Betrieb".

- Unix- und Windows-Systeme:

  Auf Unix- und Windows-Systemen ist das FT-Verwalterrecht an das Systemverwalterrecht gekoppelt, d.h.

  – Auf Unix-Systemen im Mehrbenutzerbetrieb besitzen alle Kennungen mit root-Berechtigung (UID=0) FT-Verwalterrechte.

  – Auf Unix-Systemen im Einbenutzerbetrieb hat nur die Kennung, unter der openFT abläuft, FT-Verwalterrechte.

  – Auf Windows-Systemen ist bei eingeschalteter Benutzerkontensteuerung (engl. User Account Control (UAC)) standardmäßig nur die Kennung *Administrator* ein FT-Verwalter. Andere Benutzer können openFT-Anwendungen wie den openFT Explorer jedoch mit Administrator-Rechten ausführen lassen, wenn sie beim Programmstart in dem vom Betriebssystem angezeigten Dialog die entsprechende Berechtigung erteilen.

  Nach der Neuinstallation sind FT-Verwalter und FTAC-Verwalter identisch. Das heißt, dass alle Benutzer, die auf dem System FT-Verwalterrechte besitzen, auch FTAC-Verwalter sind.

Der FTAC-Verwalter ist dadurch ausgezeichnet, dass in seinem Berechtigungssatz das entsprechende Privileg definiert ist. Wenn die Rollen von FT- und FTAC-Verwalter getrennt werden sollen, muss der FT-Verwalter den Berechtigungssatz des (künftigen) FTAC-Verwalters privilegieren. Dies ist z.B. sinnvoll, wenn der Datenschutzverantwortliche jemand anderes ist als der Systemverwalter des Rechners.

Weitere Details siehe Handbuch "openFT (Unix- und Windows-Systeme) - Installation und Betrieb".

## 7.2 Lokale Administration

Zu den Aufgaben des FT-Verwalters gehören:

- openFT starten und beenden
- openFT-Betriebsparameter verwalten
- RSA-Schlüssel verwalten
- openFT-Betrieb überwachen und steuern
- Partner verwalten
- Diagnose und Fehlerbehebung

Zu den Aufgaben des FTAC-Verwalters gehören

- Berechtigungssätze und Berechtigungsprofile verwalten

### 7.2.1 openFT starten und beenden

Die Aufgabe sieht bei Mainframe-Systemen und Unix-/Windows-Systemen unterschiedlich aus:

- Auf Mainframe-Systemen muss zuerst das openFT-Subsystem (BS2000) gestartet bzw. openFT geladen werden (z/OS). Damit stehen FT-Funktionen zur Verfügung, es werden jedoch nur synchrone Outbound-Aufträge ausgeführt, andere lokal gestellte Aufträge werden im Auftragsbuch gespeichert. Um asynchrone Aufträge ausführen zu können (Outbound und Inbound), muss der FT-Verwalter den openFT Serverprozess explizit per Kommando starten.

  Der FT-Verwalter kann den openFT Serverprozess jederzeit explizit per Kommando beenden und wieder starten.

- Auf Unix- und Windows-Systemen wird openFT normalerweise mit dem System automatisch gestartet und beendet. Der FT-Verwalter kann bei laufendem Rechner den asynchronen openFT-Server beenden (und wieder starten). Ist der asynchrone openFT-Server beendet, dann werden nur noch synchrone outbound-Aufträge ausgeführt. Andere lokal gestellte Aufträge werden im Auftragsbuch gespeichert. Diese Funktion kann z.B. benutzt werden, um den Inbound-Betrieb bei Wartungsarbeiten oder aus Sicherheitsgründen vorübergehend einzustellen.

### 7.2.2 openFT-Betriebsparameter verwalten

openFT besitzt eine Reihe von Einstellungen, die nur der FT-Verwalter festlegen und bei Bedarf ändern kann. Diese werden auch als openFT-Betriebsparameter bezeichnet. Nach der Installation sind Standardwerte gesetzt, die auf Erfahrungswerten beruhen und für den Normalbetrieb in der Regel ausreichen.
Der FT-Verwalter muss diese Einstellungen daher nur anpassen, wenn dies für das jeweilige System notwendig ist. Die nachfolgend aufgeführten Einstellungen gibt es auf allen Plattformen:

- Einstellungen für das lokale openFT-System:
  - Identifikation und den Namen des lokalen Systems
  - Adressen für die einzelnen Protokolle (openFT, FTP, FTAM)
- Performancerelevante Einstellungen:
  - Maximalwerte für den Betrieb, z.B. für die Anzahl paralleler Verbindungen, die Anzahl Aufträge im Auftragsbuch oder die Lebensdauer von Aufträgen
  - Umfang der Messdatenerfassung, siehe Seite 149
- Sicherheitseinstellungen:
  - Standardwert für die Sicherheitsstufe von Partnern
  - Benutzerdatenverschlüsselung
  - Länge des verwendeten RSA-Schlüssel
  - RSA-Mindestschlüssellänge
  - AES-Mindestschlüssellänge
  - Modus für die Absenderüberprüfung
- Protokollierung-Einstellungen:
  - Umfang des Logging
  - Automatisches Löschen von Logging-Sätzen
  - Ein-/Ausschalten von Traces
  - Umschalten von Logging-Datei und Trace-Datei
  - Umfang der Traps (Konsolen-Traps, ADM-Traps und ggf. SNMP-Traps)

Einige Einstellungen sind nur auf bestimmten Plattformen verfügbar:

- Variante der verwendeten Code-Tabelle einstellen (Unix-/Windows-Systeme, z/OS)
- Einstellungen für den Fernadministrations-Server festlegen (Unix-Systeme und Windows-Systeme)
- Verwendung von TNS und CMX (Unix-Systeme und Windows-Systeme) einstellen
- Application Entity Title (AET) für den FTAM-Betrieb ein- und ausschalten (BS2000-Systeme, Unix-Systeme und Windows-Systeme)

Details finden Sie im jeweiligen openFT-Handbuch „Installation und Betrieb“.

### 7.2.3 RSA-Schlüssel verwalten

Die Verwaltung der RSA-Schlüsselpaarsätze ist Aufgabe des FT- Verwalters und besteht aus folgenden Teilaufgaben:

- RSA-Schlüsselpaarsätze für die lokale openFT-Instanz erzeugen
- Eigenschaften aller Schlüssel im lokalen System ausgeben
- die öffentlichen Schlüssel aktualisieren
- lokale RSA-Schlüsselpaare löschen
- RSA-Schlüsselattribute modifizieren
- RSA-Schlüssel importieren

### 7.2.4 openFT-Betrieb überwachen und steuern

Die Überwachung und Steuerung des openFT-Betriebs besteht im Wesentlichen aus folgenden Teilaufgaben:

- Aufträge verwalten
- FT-Logging verwalten
- Administration über SNMP
- Messdatenerfassung steuern
- Konsolmeldungen auswerten
- ADM-Traps auswerten

#### 7.2.4.1 Aufträge verwalten

Im Auftragsbuch werden alle asynchronen Outbound-Aufträge sowie alle Inbound-Aufträge gespeichert. Der FT-Verwalter kann sich über alle noch nicht abgeschlossenen asynchronen Aufträge auf Ihrem Rechner informieren. Dazu gehört auch das Recht, Informationen über Aufträge aller Benutzer abzufragen.

Der FT-Verwalter kann

- die Bearbeitungsreihenfolge von Aufträgen des lokalen Systems ändern. Die kann z.B. notwendig sein, wenn bei hoher Last (volles Auftragsbuch) ein dringender Übertragungsauftrag sofort erledigt werden muss.

- asynchrone Aufträge des lokalen Systems löschen. Dies ist z.B. nötig, wenn Aufträge an einen Partner gestellt wurden, dessen Adresse sich geändert hat, und die Benutzer, die die Aufträge erteilt hatten, nicht erreichbar sind. Der FT-Verwalter muss natürlich die Benutzer informieren.

#### 7.2.4.2 FT-Logging verwalten

openFT kann das Ergebnis aller Dateiübertragungsaufträge protokollieren, unabhängig davon, ob die Initiative im lokalen (Outbound-Auftrag) oder fernen System (Inbound-Auftrag) liegt. Die Informationen zu einem abgeschlossenen oder abgebrochenen Auftrag werden in einem so genannten „FT-Logging-Satz“ festgehalten. Dadurch kann der gesamte FT-Betrieb lückenlos, auch über längere Zeiträume hinweg, dokumentiert werden.

Bei Einsatz von FTAC wird immer auch ein FTAC-Logging durchgeführt. Die FTAC-Funktionalität ist auf Unix- und Windows-Systemen immer installiert und aktiviert, auf Mainframe-Systemen ist FTAC optional.

Wenn zentrale Administration eingesetzt wird, ist auch ein ADM-Logging für die zentrale Administration verfügbar.

Der FT-Verwalter hat folgende Aufgaben

- Umfang des Logging einstellen über Betriebsparameter

  Der Logging-Umfang lässt sich z.B. einschränken auf Aufträge, bei denen Fehler aufgetreten sind. Das FT-Logging lässt sich komplett ausschalten, was im Sinne einer lückenlosen Protokollierung nicht zu empfehlen ist.

- Logging-Datei umschalten (offline-Logging)
- Logging-Sätze sichern und ggf. löschen
- Logging-Sätze zur Diagnose nutzen, siehe Abschnitt „Diagnose und Fehlerbehebung“ auf Seite 153.

##### Logging-Datei umschalten und Offline-Logging verwalten

Der FT- Verwalter kann die Logging-Datei über eine Betriebsparameter-Einstellung umschalten. Damit wird die aktuelle Logging-Datei geschlossen, bleibt aber als Offline-Logging-Datei erhalten. Für die folgenden Logging-Sätze wird eine neue Logging-Datei mit aktuellem Datum im Suffix erzeugt. Die Logging-Datei lässt sich mehrmals umschalten, damit lassen sich mehrere Offline-Logging-Dateien führen. Das Umschalten hat folgende Vorteile:

- Beschleunigte Logging-Zugriffe durch kleinere Logging-Datei.
- Bessere Verwaltbarkeit der Logging-Sätze durch regelmäßiges Umschalten und Sichern der Offline-Logging-Dateien.

- Möglichkeit einer umfangreichen Offline-Logging-Recherche ohne Beeinflussung des laufenden openFT-Betriebs.

#### Logging-Sätze sichern und löschen

Abhängig vom Auftragsvolumen sollte der FT-Verwalter in regelmäßigen Zeitabständen die Logging-Sätze aus der aktuellen Logging-Datei oder aus der/den Offline-Logging-Datei(en) sicherstellen, beispielsweise als Datei in CSV-Format und anschließend diese Logging-Sätze oder Offline-Logging-Datei(en) löschen. Dadurch bleiben zum einen die Logging-Sätze für eine lückenlose Dokumentation über einen längeren Zeitraum erhalten, zum anderen wird nicht unnötig Speicherplatz belegt.

Auf BS2000- und z/OS-Systemen ist zu beachten, dass sich durch das Löschen von Logging-Sätzen die zugewiesene Dateigröße der aktuellen Logging-Datei nicht ändert, sondern nur nicht mehr benötigter Platz innerhalb der Datei freigegeben wird.

#### Automatisches Löschen von Logging-Sätzen

Der FT-Verwalter kann Intervalle für das automatische Löschen von Logging-Sätzen festlegen. Damit werden Logging-Sätze ab einem festgelegten Mindestalter in regelmäßigen Abständen zu einer bestimmten Uhrzeit gelöscht.

Diese automatische Löschfunktion ist nur dann aktiv, wenn openFT gestartet ist. Ist openFT zu einem vorgesehenen Löschtermin nicht gestartet, so wird der Löschauftrag beim nächsten Start nicht nachgeholt.

Nach der Installation ist das automatische Löschen von Logging-Sätzen ausgeschaltet. Der FT-Verwalter sollte diese Funktion nur einschalten, wenn das lückenlose Protokollieren von Logging-Sätzen nicht notwendig ist.

### 7.2.4.3 Administration über SNMP

SNMP steht für **S**imple **N**etwork **M**anagement **P**rotocol und wurde als Protokoll für Netzmanagement-Dienste in TCP/IP-Netzen entwickelt. openFT ermöglicht Ihnen die zentrale Überwachung und Verwaltung eines oder mehrerer openFT-Systeme mittels einer zentralen Management-Station mit grafischer Oberfläche.

Zur Unterstützung einer automatischen Überwachung meldet openFT bestimmte Ereignisse, die keine unmittelbare Reaktion auf eine Benutzereingabe sind, mit einer Konsolmeldung. Aus Konsolmeldungen können auch SNMP-Traps erzeugt werden, so dass eine FT-Überwachung mittels SNMP durchgeführt werden kann.

Auf BS2000-Systemen ist es bei Einsatz des Filetransfer-Subagenten möglich, dass SNMP-Traps auch von openFT selbst erzeugt werden (ohne den Umweg über Konsolenmeldungen).

**Voraussetzungen**

Das SNMP-basierte openFT-Management benötigt bestimmte Produkte und Komponenten. Daher muss auf den verschiedenen Plattformen Folgendes zusätzlich installiert oder aktiviert werden:

- BS2000-Systeme: Die Produkte SNMP Management ≥ V6.0, SNMP-Basic-Agent BS2000 V6.0 (SBA-BS2) und SNMP-Standard-Collection BS2000 V6.0 (SSC-BS2) müssen installiert sein.
- Windows-Systeme: Der SNMP-Dienst von Microsoft muss vorhanden sein. Der openFT-Subagent ist in openFT enthalten und wird während der Installation eingerichtet.

Beachten Sie bitte für detaillierte Informationen die entsprechenden Handbücher.

**Funktionsumfang des Filetransfer-Subagenten**

Der Filetransfer-Subagent dient

– zum Starten und Stoppen von openFT (BS2000)
– zur Informationsbeschaffung über Systemparameter
– zum Ändern des Public-Key zur Verschlüsselung
– zur Ausgabe von Statistikdaten
– zur Steuerung der Diagnose
– zur Ausgabe von Partner-Informationen

Die MIB zu openFT bietet Objekte zu den oben genannten Management-Aufgaben. Die Objekte zum Starten und Stoppen, zur Verschlüsselung des Public-Key und zur Steuerung der Diagnose bieten auch schreibende Zugriffe.

Details finden Sie in den jeweiligen openFT-Handbüchern „Installation und Betrieb“.

#### 7.2.4.4 Messdatenerfassung steuern

openFT bietet die Möglichkeit, eine Reihe von charakteristischen Daten des openFT-Betriebs zu messen und aufzuzeichnen. Die Daten lassen sich in drei Gruppen einteilen:

- Durchsatz, z.B gesamter durch openFT bedingter Netzdurchsatz
- Zeitdauer, z.B. Bearbeitungsdauer für asynchrone Aufträge
- Status, z.B. Anzahl der aktuell wartenden Aufträge

Der FT- Verwalter kann die Messdatenerfassung einschalten, um sich ein Bild über die Art der Auslastung zu machen. Die Messdatenerfassung sollte aber nicht ständig aktiviert sein, da dies die Performance beeinträchtigt.

Der FT-Verwalter hat folgende Möglichkeiten:

- Erfassung einschalten und ausschalten
- Erfassung nach Partnertyp selektieren
- Erfassung nach Auftragstyp selektieren

Die einmal gewählten Einstellungen bleiben solange erhalten, bis der FT-Verwalter sie explizit ändert. D.h. sie werden auch durch einen Neustart des Rechners nicht verändert.

### Messdaten anzeigen

Wenn die Messdatenerfassung eingeschaltet ist, können die Messdaten am lokalen System oder von einem fernen System abgerufen werden.

Im Prinzip kann jeder Benutzer die Messdaten lesen, aber in der Praxis ist es Sache des FT-Verwalters, die Daten auszuwerten, da er als einziger auch den Gesamtüberblick über das System besitzt und die Betriebsparameter ändern kann.

Die Messwerte werden in Tabellenform ausgegeben.

Auf Unix- und Windows-Systemen lassen sich die Messdaten auch über die grafische Oberfläche des openFT Monitor anzeigen. Der openFT Monitor ist „remote“-fähig, d.h. er kann auch Messwerte von fernen Systemen anzeigen. Dies kann man sich für Mainframe-Systeme wie BS2000- oder z/OS-Systeme wie folgt zunutze machen:

1. Der FT-Verwalter auf dem Mainframe-System richtet ein spezielles Berechtigungsprofil ein, das beim Aufruf des openFT Monitors angegeben wird und bewirkt, dass nur die Messwerte gelesen und übertragen werden. Das Berechtigungsprofil verwendet das Schlüsselwort *FTMONITOR als Vorverarbeitungskommando. Als Zugangsberechtigung kann z.B. HOSTMONITOR angegeben werden.
2. Auf dem Unix- oder Windows-System gibt man beim Aufruf des FT Monitors das Mainframe-System mit der Zugangsberechtigung HOSTMONITOR an. Der FT-Monitor zeigt die Mainframe-Messdaten in grafischer Form an.

Auf Windows-Systemen können die Messdaten auch über die Windows-Leistungsüberwachung angezeigt werden.

#### 7.2.4.5 Konsolmeldungen auswerten

Zur Unterstützung einer automatischen Überwachung meldet openFT weitere Ereignisse, die nicht unmittelbare Reaktion auf eine Benutzereingabe sind, mit einer Konsolmeldung.

Typische Ereignisse sind die Statusänderungen von openFT (Start/Stop) und Partnern oder fehlgeschlagene Aufträge.

Der FT-Verwalter kann per Betriebsparameter-Einstellung (Konsol-Traps) angeben, welche Ereignisse gemeldet werden.

#### 7.2.4.6 ADM-Traps auswerten

ADM-Traps sind kurze Meldungen, die openFT bei bestimmten Ereignissen, die während des openFT-Betriebs eintreten, an einen so genannten ADM-Trap-Server schickt. Zu diesen Ereignissen können z.B. fehlerhafte FT-Aufträge, Statuswechsel oder Nichterreichbarkeit von Partnern gehören.

Der FT-Verwalter kann per Betriebsparameter-Einstellung (ADM-Traps) angeben, welche Ereignisse gemeldet werden.

Weitere Details siehe Abschnitt „ADM-Traps“ auf Seite 164.

#### 7.2.4.7 Sonstige Administrationsfunktionen

Die folgenden Administrationsfunktionen sind nicht auf allen Plattformen vorhanden.

##### Jobvariable auf BS2000-Systemen

Eine openFT-Instanz lässt sich über eine von openFT automatisch versorgte MONJV überwachen. Die Jobvariable wird beim ersten Start von openFT angelegt und von da an verwendet.

##### Jobprotokoll auf z/OS

Neben der Logging-Datei enthält auch das Job-Protokoll von openFT Informationen, die für den FT-Verwalter nützlich sein können. Zum einen werden manche Meldungen nur in das Job-Protokoll von openFT ausgegeben, zum anderen ist oft die chronologische Ordnung der im Job-Protokoll enthaltenen Meldungen hilfreich bei der Diagnose von Störungen des FT-Betriebs.

### 7.2.5 Partner verwalten

Der FT-Verwalter kann steuern, mit welchen Partnern Dateiübertragung und Dateimanagement möglich ist und welche Sicherheitseinstellungen für die jeweiligen Partner gelten. Der FT-Verwalter ist für die Pflege der Partnerliste verantwortlich, d.h. er kann:

- Partner in die Partnerliste eintragen

  Dabei kann der FT-Verwalter dem Partner individuelle Eigenschaften zuordnen wie z.B.

  – eine Sicherheitsstufe
  – Authentifizierungspflicht
  – eine Prioritätsstufe
  – Trace für diesen Partner ein-/ausschalten
  – Parallele Bearbeitung von Aufträgen mit diesem Partner erlauben/verbieten
  – Wiederanlauf für Outbound-Aufträge ein- und ausschalten
  – Status-Einstellungen wie z.B. automatische Deaktivierung nach mehreren fehlerhaften Verbindungsaufbau-Versuchen

- Eigenschaften der Partner in der Partnerliste ändern wenn nötig
- Partner aus der Partnerliste löschen
- Dynamische Partner zulassen oder sperren

  Damit kann der FT-Verwalter z.B. verhindern, dass unbekannte Partner aus dem Internet per openFT auf das lokale System zugreifen.

- Partnerliste exportieren

  Damit kann der FT-Verwalter die Einträge in der Partnerliste per Kommando ausgeben und in eine Datei exportieren, z.B. um die Einträge zu sichern oder für andere Systeme zu verwenden. Bei diesem Kommando kann angegeben werden, für welche Plattform die Kommandos erzeugt werden, d.h man kann z.B. eine BS2000- Partnerliste im Unix- oder Windows-Format exportieren. Beim Exportieren werden die Einträge in entsprechende Kommandos umgesetzt, die auf dem jeweiligen System nur noch eingelesen werden müssen.

Weitere Details zu Partnerliste und Partnertypen finden Sie im Kapitel „Partnerkonzept“ auf Seite 51.

### 7.2.6 Diagnose und Fehlerbehebung

Zur Diagnose stehen dem FT-Verwalter zwei wichtige Instrumente zur Verfügung:

- Logging-Sätze

  Tritt ein ernsthaftes Problem auf, sollte der FT-Verwalter als erstes versuchen, den oder die zugehörigen FT- und ggf. FTAC-Logging-Sätze zu ermitteln. Die Logging-Sätze enthalten einen Returncode, der eine erste Auskunft über die Ursache eines Problems geben kann. In einem Logging-Satz sind außerdem alle wichtigen Parameter eines FT-Auftrags protokolliert.

- openFT-Traces

  Reichen die Logging-Sätze nicht zur Diagnose aus, kann der FT-Verwalter einen FT-Trace einschalten. Der Trace kann global oder spezifisch eingeschaltet werden, bis hin zum Trace, der nur einen bestimmten Partner betrifft.

  Die FT-Trace-Dateien können anschließend mit Hilfe eines openFT-Tools (*fttrace*) aufbereitet und ausgewertet werden.

  Auf Unix- und Windows-Systemen kann der FT-Verwalter die Trace-Dateien mit dem openFT Explorer direkt anschauen.

- Traces für untere Protokollschichten

  Auf Unix- und Windows-Systemen kann der FT-Verwalter Traces für die unteren Protokollschichten einschalten. Diese Informationen werden in den openFT-Trace geschrieben.

## 7.2.7 Berechtigungssätze und Berechtigungsprofile verwalten

Die Verwaltung von Berechtigungssätzen und Berechtigungsprofilen ist Aufgabe des FTAC-Verwalters. Die Aufgabe besteht aus folgenden Teilen:

- Standardberechtigungssatz festlegen
- Individuelle Berechtigungssätze festlegen und pflegen
- Berechtigungsprofile verwalten
- FTAC-Umgebung übertragen

### 7.2.7.1 Standardberechtigungssatz festlegen

Der FTAC-Verwalter muss zunächst ein mittleres Schutzbedürfnis für die Benutzerkennungen in seinem System ermitteln und mit dieser Information den Standardberechtigungssatz ändern. Im Standardberechtigungssatz werden Festlegungen für den „durchschnittlichen“ FTAC-Benutzer im jeweiligen System getroffen. Sie bieten für die meisten Benutzer ausreichenden Schutz. Außerdem kann in jedem Berechtigungssatz an verschiedenen Stellen auf den Standardberechtigungssatz Bezug genommen werden. Das hat den Vorteil, dass eventuelle Änderungen des Standardberechtigungssatzes automatisch in diese Berechtigungssätze übernommen werden.

Für Benutzerkennungen, deren Schutzbedürfnis vom Durchschnitt abweicht, kann der FTAC-Verwalter individuelle Festlegungen treffen.

### 7.2.7.2 Individuelle Berechtigungssätze festlegen und pflegen

Der FTAC-Verwalter kann für jede Benutzerkennung einen individuellen Berechtigungssatz festlegen. Dies ist z.B. sinnvoll, wenn der Standardberechtigungssatz aus Sicherheitsgründen sehr restriktiv eingestellt ist, z.B. kein Inbound-Zugriff erlaubt (Inbound Senden und Empfangen sind auf 0 gesetzt).

Wenn für eine Kennung trotzdem Inbound-Übertragung erlaubt sein soll (d.h. mit Initiative von außerhalb), dann legt der FTAC-Verwalter für diese Kennung einen individuellen Berechtigungssatz fest, indem er die Sicherheitsstufe für Inbound Senden/Empfangen hochsetzt. Der Benutzer muss seine Einstellung ggf. dann ebenfalls hochsetzen, da in seiner Einstellung evtl. noch die ursprüngliche, restriktive Vorgabe des Standardberechtigungssatzes steht.

### 7.2.7.3 Berechtigungsprofile verwalten

Der FTAC-Verwalter hat folgende Möglichkeiten, fremde Berechtigungsprofile zu bearbeiten:

- Er kann Berechtigungsprofile für fremde Benutzerkennungen erstellen; dabei bestehen aber bestimmte Einschränkungen.
- Er kann sie sich ansehen. Die Zugangsberechtigung eines Berechtigungsprofils wird nicht mit ausgegeben. Das heißt, dass der FTAC-Verwalter durch seine Verwaltereigenschaft keine Zugriffsrechte auf die Dateien fremder Benutzerkennungen bekommt.
- Er kann sie löschen. Diese radikalste aller Möglichkeiten sollte aber nur in begründeten Ausnahmefällen und nach Rücksprache mit dem Eigentümer des Profils angewendet werden.
- Er kann sie privilegieren (siehe Seite 156), oder eine Privilegierung wieder zurücknehmen.
- Er kann auch andere Änderungen vornehmen. Falls der FTAC-Verwalter weder die nötigen Systemverwalterrechte besitzt, noch die komplette LOGON/Login-Berechtigung des Profileigentümers angibt, wird das Berechtigungsprofil dann so lange gesperrt, bis der Profileigentümer diese Änderungen quittiert, indem er die Zugangsberechtigung wieder auf "gültig" setzt.

#### Anlegen von Berechtigungsprofilen für fremde Benutzerkennungen

Wenn der FTAC-Verwalter ein Berechtigungsprofil für eine fremde Benutzerkennung anlegen will, gibt es mehrere Vorgehensweisen:

- Wenn der FTAC-Verwalter die notwendigen Administrationsrechte besitzt (siehe Abschnitt „Rollenkonzept für die Administration“ auf Seite 141), dann darf er uneingeschränkt Berechtigungsprofile für andere Kennungen einrichten, auch ohne Kenntnis des aktuellen Benutzerkennwortes. In diesen Profilen darf der FTAC-Verwalter eine Zugangsberechtigung angeben, die sofort nach dem Einrichten in FT-Aufträgen verwendet werden kann. Beachten Sie, dass sich FTAC-Verwalter mit diesen Administrationsrechten durch Einrichten entsprechender Berechtigungsprofile Zugang zu den Dateien jeder beliebigen Benutzerkennung verschaffen und dadurch eventuell Schutzvorschriften umgehen können!

  Die notwendigen Administrationsrechte hängen von der Plattform ab:
  – BS2000-Systeme: TSOS-Privileg
  – z/OS-Systeme: SU-Privileg
  – Unix- und Windows-Systeme: FTAC-Verwalterrechte

- Wenn der FTAC-Verwalter die notwendigen Administrationsrechte **nicht** besitzt, gibt es zwei Möglichkeiten:
  - Er gibt die vollständige LOGON-/login-Berechtigung an (d.h. Benutzerkennung, Kennwort und ggf. Abrechnungsnummer). Dann kann er auch eine Zugangsberechtigung angeben. Er legt damit ein gültiges Berechtigungsprofil an, d.h. dieses Profil kann sofort in File-Transfer- und File-Management-Aufträgen verwendet werden.

    In einem solchen Berechtigungsprofil ist aber das Kennwort des Benutzers fest gespeichert. Wenn der Benutzer sein Kennwort ändert, muss also auch das Berechtigungsprofil geändert werden.
  - Er gibt nur die Benutzerkennung an (ohne Passwort und ggf. Abrechnungsnummer). Dann wird das Profil ohne Zugangsberechtigung angelegt; diese muss der Benutzer später selber vergeben.

**Berechtigungsprofile privilegieren**

Das Vorgehen beim Privilegieren eines Berechtigungsprofils ist einfach:

1. Der Benutzer richtet ein Berechtigungsprofil für die geplante Aufgabe ein.
2. Der FTAC-Verwalter schaut sich das Berechtigungsprofil an, um festzustellen, ob das Profil eine Gefährdung für den Datenschutz darstellt.
3. Wenn das Profil unbedenklich ist, privilegiert es der FTAC-Verwalter.

### 7.2.7.4 FTAC-Umgebung übertragen - Die Environment-Funktionen

Der FTAC-Verwalter kann Berechtigungsprofile und -sätze in eine Datei schreiben lassen („exportieren“). Damit kann er eine Sicherung von allen Berechtigungsprofilen und -sätzen anlegen, die auf dem Rechner existieren.

Außerdem ist diese Funktion sinnvoll, falls ein Benutzer auf einen anderen Rechner umzieht. Der FTAC-Verwalter sichert in diesem Fall zunächst die bisherige FTAC-Umgebung in einer Datei, anschließend spielt er sie wieder auf dem anderen Rechner ein. Der FTAC-Benutzer kann dann mit der gleichen FTAC-Umgebung arbeiten wie bisher, also den gleichen Berechtigungsprofilen und dem gleichen Berechtigungssatz.

Wenn der FTAC-Verwalter nicht die nötigen Administrationsrechte besitzt (siehe Seite 155), müssen eventuell vorhandene Privilegien auf dem neuen Rechner explizit wieder eingerichtet und die Zugangsberechtigungen explizit freigegeben werden. Besitzt der FTAC-Verwalter dagegen die nötigen Administrationsrechte, dann kann er beim Importieren angeben, ob die Profile mit unveränderten Attributen übernommen werden oder nicht.

Der FTAC-Verwalter hat die Möglichkeit, durch die FTAC-Umgebung entsprechende Parameterangaben gezielt zu sichern und wieder herzustellen:

– Berechtigungsprofile und Berechtigungssätze eines Benutzers oder mehrerer Benutzer (bis zu 100 Benutzer)
– alle Berechtigungsprofile und Berechtigungssätze, die auf dem Rechner vorhanden sind
– nur Berechtigungssätze und keine Berechtigungsprofile
– nur Berechtigungsprofile und keine Berechtigungssätze

Den Inhalt einer Sicherungsdatei kann der FTAC-Verwalter sich anzeigen lassen.

## 7.3 Remote Administration über openFT Explorer

Einen Teil Administrationstätigkeiten, die im lokalen System zur Verfügung stehen, können Sie über den openFT Explorer auch auf fernen openFT-Systemen duchführen. Damit lassen sich ferne openFT-Systeme auf beliebigen Plattformen auf sehr einfache Art administrieren. Der Funktionsumfang umfasst folgenden Teil der lokalen Administration:

- openFT-Betriebsparameter verwalten

  Damit können Sie für das ferne System alle Betriebsparameter einstellen, siehe Abschnitt „openFT-Betriebsparameter verwalten“ auf Seite 145.

- openFT-Monitor starten

  Sie können den openFT-Monitor auf dem fernen System starten, siehe Abschnitt „Messdatenerfassung steuern“ auf Seite 149. Dazu muss das openFT-Monitoring eingeschaltet sein, z.B. per Betriebsparameter über Remote Administration.

Diese Funktionen stehen im openFT Explorer für alle Partner zur Verfügung, die im Knoten **Partner** definiert sind. Dabei müssen Sie die Zugangsberechtigung für das ferne System hinterlegen. Wenn Sie den vollen Funktionsumfang nutzen möchten, muss die angegebene Zugangsberechtigung die FT-Administrationsberechtigung für das ferne System besitzen.

## 7.4 Zentrale Administration

Die zentrale Administration von openFT umfasst die Funktionen **Fernadministration** und **ADM-Traps**. Auf Unix- und Windows-Systemen unterstützt openFT beide Funktionen in vollem Umfang.

Diese Funktionen bieten erhebliche Vorteile, wenn Sie eine größere Anzahl von openFT-Instanzen administrieren und überwachen möchten, z.B.:

- Einfaches Konfigurieren

  Die Konfigurationsdaten werden zentral auf dem **Fernadministrations-Server** gehalten und sind dadurch nur einmal vorhanden. Rollenbildung in Form von **Fernadministratoren** und Gruppierung mehrerer Instanzen erlauben es, auch komplexe Konfigurationen einfach und übersichtlich zu realisieren. Spätere Änderungen sind leicht einzubringen und machen die Konfiguration damit wartungsfreundlich.

  Der Fernadministrations-Server läuft auf einem Unix- oder Windows-System.

- Einfacheres Authentifizierungsverfahren

  Wenn Sie aus Sicherheitsgründen mit Authentifizierung arbeiten möchten, dann müssen nur wenige öffentliche Schlüssel verteilt werden:

  – Für die Strecke zum Fernadministrations-Server müssen die Schlüssel der Rechner, von denen aus administriert werden soll, auf dem Fernadministrations-Server hinterlegt werden.

  – Für die Strecke vom Fernadministrations-Server zu den zu administrierenden Instanzen muss nur der öffentliche Schlüssel des Fernadministrations-Servers auf den zu administrierenden openFT-Instanzen hinterlegt werden.

- Hohe Leistungsfähigkeit

  Die Fernadministrations-Schnittstelle ermöglicht wesentlich längere Kommando-Sequenzen als in openFT bis V10.0.

  Außerdem kann der Fernadministrations-Server so konfiguriert werden, dass er ausschließlich für die zentrale Administration zur Verfügung steht. In diesem Falle gibt es keine Abhängigkeiten zum normalen FT-Betrieb und damit auch keine gegenseitige Beeinträchtigung.

- Einfaches Administrieren

  Fernadministratoren benötigen nur eine (zentrale) Zugangsberechtigung. Ohne zentrale Administration müssen sich Fernadministratoren die Zugangsdaten von jeder zu administrierenden openFT-Instanz merken.

- Zentrale Protokollierung wichtiger Ereignisse

  Bei bestimmten Ereignissen auf openFT-Instanzen können ADM-Traps erzeugt werden, die an den (zentralen) ADM-Trap-Server geschickt und dort dauerhaft gespeichert werden. Damit haben Fernadministratoren die Möglichkeit, wichtige Ereignisse auch nachträglich und instanzspezifisch auszuwerten.

- Kompatible Integration früherer openFT-Versionen

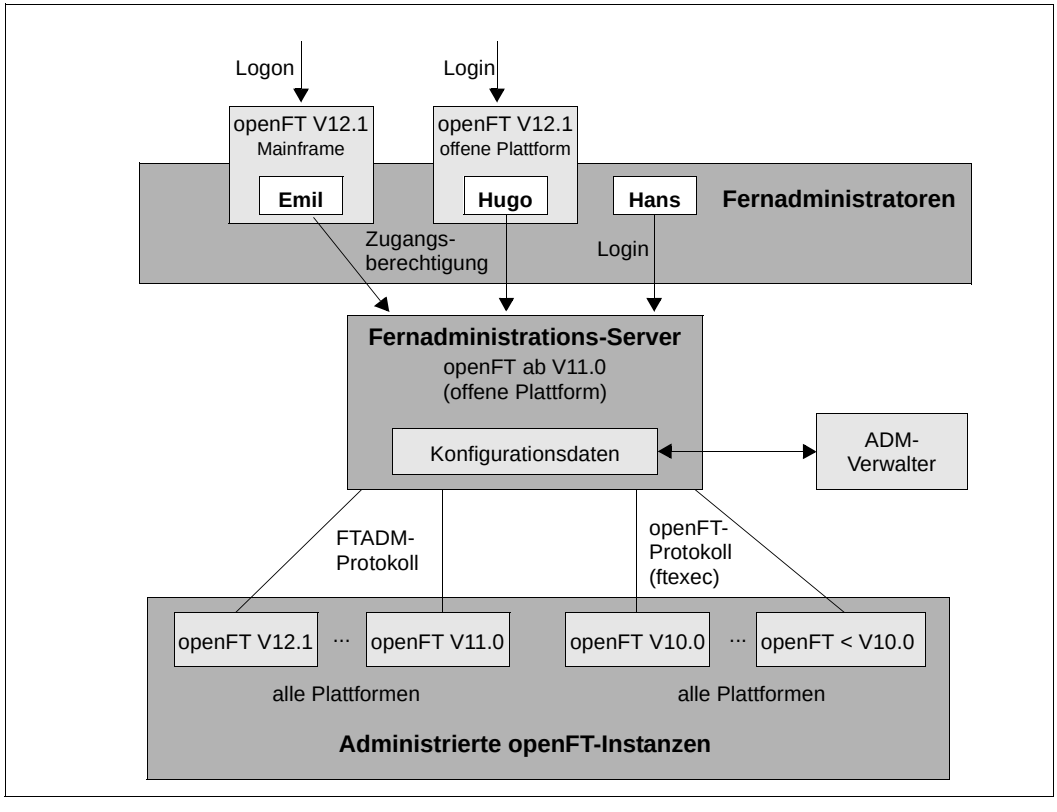
  Instanzen mit openFT-Versionen ab V8.0 können einfach in die Konfiguration aufgenommen werden und auf dieselbe Art und Weise administriert werden wie Instanzen ab V11.0. Dabei lassen sich alle Administrationsfunktionen nutzen, die die jeweilige openFT-Version bietet.

## 7.4.1 Fernadministration

Mit openFT können Sie einen Fernadministrations-Server einrichten, über den Sie Ihre openFT-Instanzen auf den unterschiedlichen Plattformen administrieren können.
Als Administrations-Arbeitsplatz können Sie eine beliebige openFT-Instanz wählen.

### 7.4.1.1 Konzept der Fernadministration

Das folgende Bild zeigt anhand eines Einsatz-Szenarios die Komponenten der Fernadministration und die wichtigsten Konfigurationsmöglichkeiten.




Komponenten der Fernadministration

Die Fernadministration besteht aus folgenden Komponenten:

**Fernadministrations-Server**

Zentrale Komponente der Fernadministration. Diese läuft auf einem Unix- oder Windows-System mit openFT ab V11.0 und enthält sämtliche Konfigurationsdaten für die Fernadministration.

In einer Gesamtkonfiguration können mehrere Fernadministrations-Server definiert werden, siehe Seite 162.

**ADM-Verwalter**

Person, die den Fernadministrations-Server verwaltet. Sie erstellt die Konfigurationsdaten für die Fernadministration, in denen z.B. die Fernadministratoren und die administrierten openFT-Instanzen festgelegt sind. Der ADM-Verwalter ist die einzige Person, die die Konfigurationsdaten ändern darf.

Zum einfachen Erstellen und Editieren der Konfigurationsdaten kann der ADM-Verwalter die grafísche Oberfläches des KJonfigurations-Editors benutzen.

**Fernadministrator**

Rolle, die im Fernadministrations-Server konfiguriert wird und dazu berechtigt, bestimmte Administrationsfunktionen auf bestimmten openFT-Instanzen auszuführen. Ein Fernadministrator kann sich

– direkt am Fernadministrations-Server anmelden (Single Sign-on)

– an eine andere openFT-Instanz (ab V11.0) anmelden und mittels FTAC-Zugangsberechtigung auf den Fernadministrations-Server zugreifen.
  Die openFT-Instanz kann sowohl auf Mainframes (BS2000, z/OS) als auch auf Unix- oder Windows-Systemen ablaufen. Für die Kommunikation wird das FTADM-Protokoll verwendet.

Es können mehrere Fernadministratoren mit unterschiedlichen Rechten konfiguriert werden.

**Administrierte openFT-Instanz**

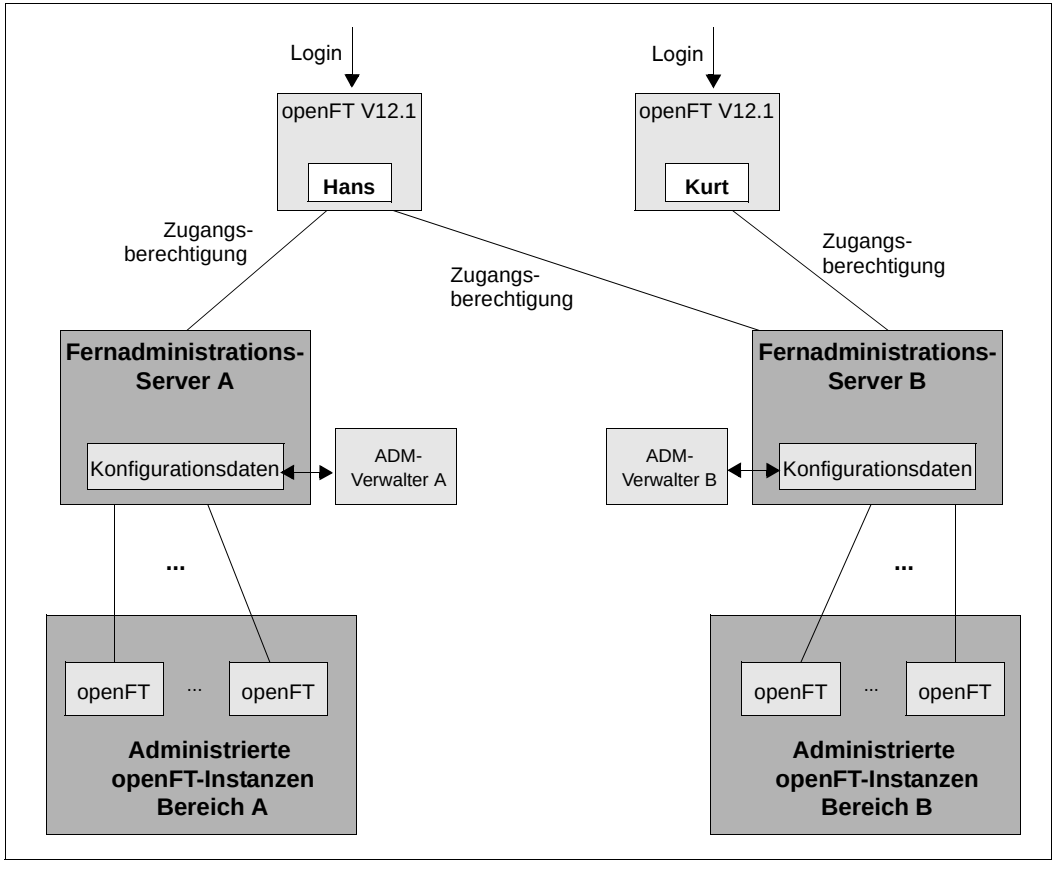
openFT-Instanz, die durch Fernadministratoren im laufenden Betrieb administriert werden kann. Der Zugriff erfolgt über ein Berechtigungsprofil. Je nachdem, welche openFT-Version die openFT-Instanz besitzt, gilt:

– Bei openFT-Instanzen ab V11.0 wird das FTADM-Protokoll verwendet, es kann der volle Funktionsumfang der Fernadministration genutzt werden.

– Bei openFT-Instanzen von V8.0 bis V10.0 wird die Administration über das openFT-Protokoll und das Kommando *ftexec* durchgeführt. Der Funktionsumfang richtet sich nach der openFT-Version der administrierten Instanz.

#### 7.4.1.2 Konfiguration mit mehreren Fernadministrations-Servern

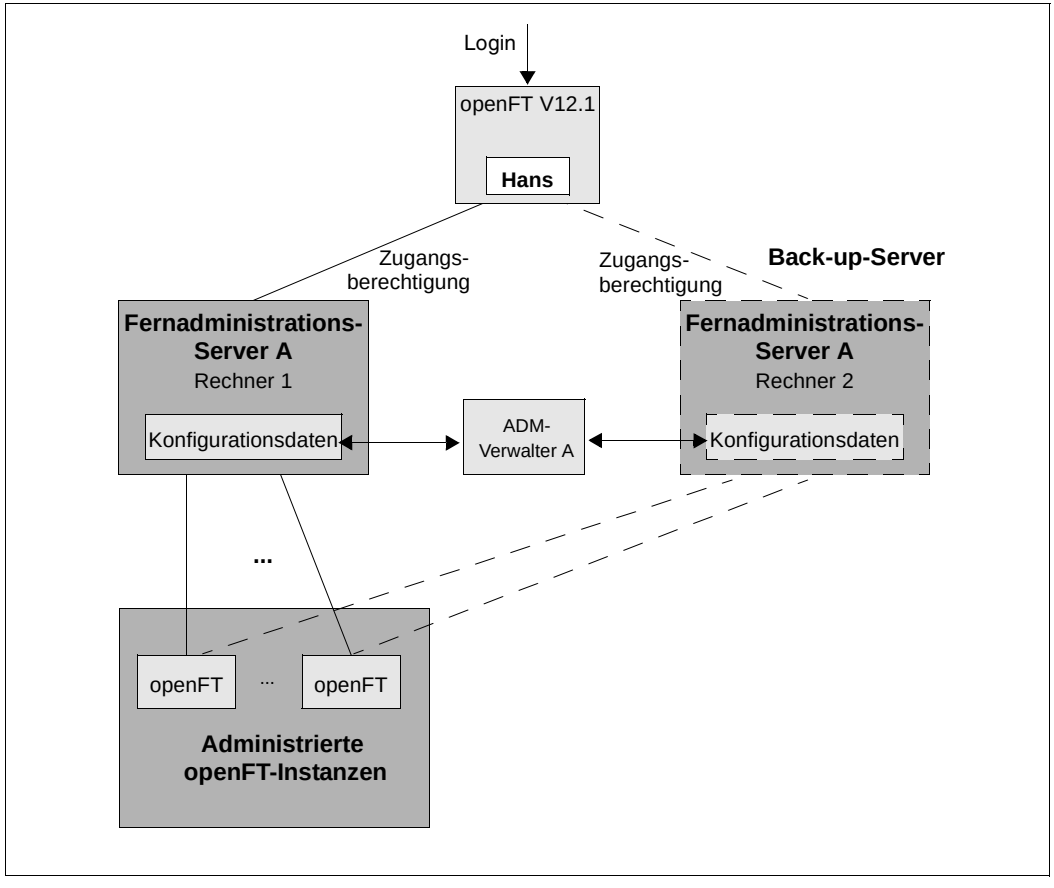
Es lassen sich auch komplexe Konfigurationen definieren, bei denen Fernadministratoren auf mehrere Fernadministrations-Server zugreifen. Das folgende Diagramm zeigt dies an einem Beispiel.




Getrennte Konfigurationen mit zwei Fernadministrations-Servern

Bereich A und B sind zwar strikt getrennt, aber *Hans* darf Instanzen aus den Bereichen A und B administrieren, *Kurt* dagegen nur aus dem Bereich B.

Dieselbe Methode kann auch dazu verwendet werden, eine redundante Konfiguration mit einem zweiten Fernadministrations-Server zu definieren. Damit lässt sich eine einfache Back-up-Lösung realisieren.




Redundante Konfiguration mit zweitem Fernadministrations-Server als Back-up

Bei Ausfall von Rechner 1 kann der Fernadministrator den Rechner 2 als Fernadministrations-Server nutzen. Voraussetzung ist, dass

– der ADM-Verwalter die Konfigurationsdaten auf beiden Rechnern immer konsistent hält,
– die Berechtigungsprofile für den Zugang zum Fernadministrations-Server sowie die Partnerlisteneinträge (falls verwendet) auf Rechner 1 und Rechner 2 identisch sind,
– auf den administrierten Instanzen die Berechtigungsprofile so definiert sind, dass sie beide Fernadministrations-Server als Partner akzeptieren.

Wenn mit Authentifizierung gearbeitet wird, dann müssen Sie zusätzlich beachten, dass

– die Schlüssel der Rechner, von denen aus administriert wird, auf beiden Fernadministrations-Servern vorhanden sind,

– die administrierten Instanzen die Schlüssel von beiden Fernadministrations-Servern benötigen.

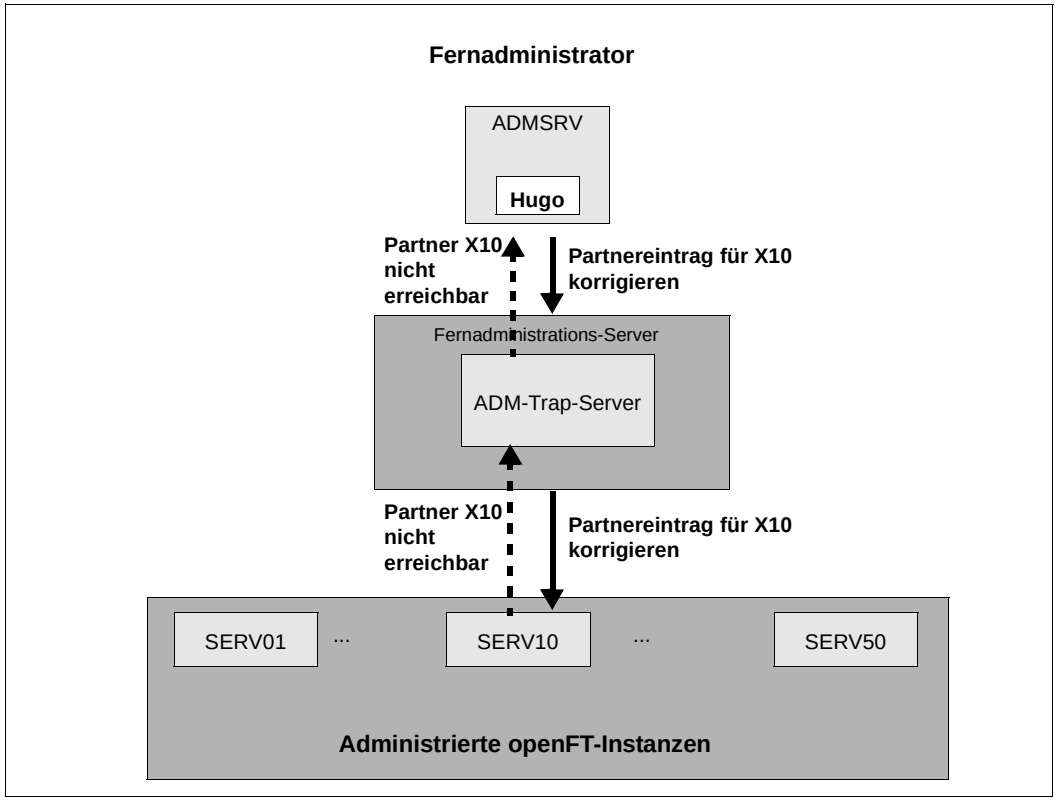
Daher sollten Sie vor allem bei komplexen Konfigurationen die Ausfallsicherheit des Fernadministrations-Servers mit Hilfe eines Clusters realisieren. Beispiele für das Einrichten
eines Clusters finden Sie im Handbuch "openFT (Unix- und Windows-Systeme) - Installation und Betrieb".

### 7.4.2 ADM-Traps

ADM-Traps sind kurze Meldungen, die openFT bei bestimmten Ereignissen, die während des openFT-Betriebs eintreten, an einen so genannten **ADM-Trap-Server** schickt. Zu diesen Ereignissen können z.B. fehlerhafte FT-Aufträge, Statuswechsel oder Nichterreichbarkeit von Partnern gehören.

ADM-Traps werden auf dem ADM-Trap-Server dauerhaft gespeichert. Damit lassen sich openFT-Systeme an zentraler Stelle überwachen. Der FT-Verwalter des ADM-Trap-Servers kann sich dabei mittels des openFT Explorers oder des Kommandos *ftshwatp* auf einfache Weise einen Überblick über Ereignisse verschaffen, die auf von ihm überwachten openFT-Instanzen aufgetreten sind.

Wenn der ADM-Trap-Server gleichzeitig auch als Fernadministrations-Server eingesetzt wird, dann können Fernadministratoren ADM-Traps auch von anderen Systemen aus einsehen und damit die Systeme überwachen, die sie administrieren, siehe folgendes Beispiel:




ADM-Trap-Server auf dem Fernadministrations-Server

SERV10 meldet per ADM-Trap, dass Partner X10 nicht erreichbar ist. Der Fernadministrator HUGO liest diesen Trap, prüft u.a. die Logging-Sätze auf SERV10 und stellt fest, dass sich die IP-Adresse von X10 geändert hatte, diese Änderung aber noch nicht in der Partnerliste nachgezogen wurde. Über den Fernadministrations-Server ändert HUGO auf SERV10 mit dem entsprechenden FT-Kommando den Partnerlisteneintrag für X10.

# 8 Lizenzrechtliche Bestimmungen

Die folgenden Bestimmungen betreffen die Nutzung von *libxml2*, Secure FTP und xerces-J für openFT-Script.

### Nutzung von libxml2

Für die Verarbeitung der XML-Daten wird die *libxml2* verwendet, die den XML C Parser und ein XML Toolkit enthält. *libxml2* wurde ursprünglich für das Gnome Project entwickelt, ist jedoch auch außerhalb von Gnome verwendbar. Es handelt sich bei *libxml2* um freie Software, die unter der MIT-Lizenz verfügbar ist:

### Nutzung von openSSL für Secure FTP

Die folgenden Bestimmungen betreffen den Betrieb mit 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
---------------

/* ====================================================================
Copyright (c) 1998-2006 The OpenSSL Project. 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. All advertising materials mentioning features or use of this software must
   display the following acknowledgment:
   "This product includes software developed by the OpenSSL Project for use in
   the OpenSSL Toolkit. (http://www.openssl.org/)"

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse
   or promote products derived from this software without prior written
   permission. For written permission, please contact openssl-core@openssl.org.

5. Products derived from this software may not be called "OpenSSL" nor may
   "OpenSSL" appear in their names without prior written permission of the
   OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following acknow-
   ledgment:
   "This product includes software developed by the OpenSSL Project for use in
   the OpenSSL Toolkit (http://www.openssl.org/)"
```

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``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 OpenSSL PROJECT 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 product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License
---

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved.

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.]

### Nutzung von xerces-J für openFT-Script

Die folgenden Bestimmungen betreffen den Betrieb mit 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.
*/
```

# Fachwörter

Mit *Kursivschrift* wird auf weitere Fachwörter verwiesen.

**ABEND**
: Abnormal Ending. Abnormale Programm-Beendigung in z/OS.

**absoluter Pfadname**
absolute path name
: Gesamtweg von der Wurzel des Dateisystems bis zur Datei.

**access control**
access control
: FTAM-spezifisches Dateiattribut im *virtuellen Dateispeicher*, gehört zur *security group*, legt *Zugriffsrechte* fest.

**ACF-2**
: Programm-Produkt der Firma Computer Associates für die Zugangs- und Zugriffssicherung auf z/OS.

**action list**
action list
: Element des FTAM-spezifischen Dateiattributs *access control* (gehört zur *security group*) im *virtuellen Dateispeicher*, legt *Zugriffsrechte* fest.

**ADM-Partner**
ADM partner
: Partnersystem einer openFT-Instanz, mit dem über das *FTADM-Protokoll* kommuniziert wird, um *Fernadministration* durchzuführen.

**ADM-Traps**
ADM traps
: Kurze Meldungen, die bei bestimmten Ereignissen, die während des openFT-Betriebs auftreten, an den *ADM-Trap-Server* gesendet werden.

**ADM-Trap-Server**
ADM trap server
: Server, der die *ADM-Traps* empfängt und dauerhaft speichert. Er muss als *Fernadministrations-Server* konfiguriert sein.

**ADM-Verwalter**
ADM administrator
: Verwalter des *Fernadministrations-Servers*. Er darf als einzige Person die Konfigurationsdaten des Fernadministrations-Servers ändern.

**Administrierte openFT-Instanzen**
administrated openFT instances
: openFT-Instanzen, die durch *Fernadministratoren* im laufenden Betrieb administriert werden können.

**AES (Advanced Encryption Standard)**
: Aktueller symmetrischer Verschlüsselungsstandard, festgelegt vom NIST (National Institute of Standards and Technology), basierend auf dem an der Universität Leuven (B) entwickelten Rijndael-Algorithmus. Das AES-Verfahren wird von der openFT-Produktfamilie zur Verschlüsselung der Auftragsbeschreibungsdaten und ggf. der Dateiinhalte verwendet.

**alphanumerisch**
alphanumeric
: Alphanumerische Zeichen umfassen alphabetische und numerische Zeichen, d.h. die Buchstaben A-Z und die Ziffern 0-9, auf z/OS zusätzlich die Sonderzeichen $, @, #.

**AMODE**
: Angabe zur Adressierung eines Moduls (24-Bit- oder 31-Bit-breite Adressen) in z/OS.

**ANSI-Code**
ANSI code
: Normierter 8-Bit-Zeichensatz für den Nachrichtenaustausch. Das Akronym steht für „American National Standards Institute“.

**API (Application Programming Interface)**
: Ein Interface, das Anwendungsprogrammierern zur freien Verfügung steht. Es bietet eine auf eine bestimmte Funktionalität ausgelegte Menge von Schnittstellenmechanismen an.

**Application Entity Title (AET)**

Der Application Entity Title ist eine Schicht 7-Adress-Information des *ISO-Referenzmodells*. Er ist nur für *FTAM-Partner* von Bedeutung.

**asynchroner Auftrag**

asynchronous request

Der *FT-Auftrag* wird nach der Auftragsabgabe entkoppelt vom Benutzer durchgeführt. Der Benutzer kann weiterarbeiten, nachdem das System die Annahme des Auftrags bestätigt hat. (vgl. *synchroner Auftrag*)

**Auftrag**

request

Siehe *FT-Auftrag*

**Auftragsbuch**

request file

Datei, die *asynchrone Aufträge* und ihre Bearbeitungszustände enthält.

**Auftrags-Identifikation**

request identification / request ID

Vom lokalen System vergebene Nummer zur Identifikation eines *FT-Auftrags*.

**Auftragsnummer**

request number

siehe *Auftrags-Identifikation*.

**Auftragsverwaltung**

request management

FT-Funktion, die *FT-Aufträge* verwaltet und dafür sorgt, dass sie von der Abgabe des Auftrags bis zur Erledigung bzw. Beendigung bearbeitet werden.

**Authentifizierung**

authentication

Verfahren, mit dem openFT die eindeutige Identität des Auftragspartners überprüft.

**Benannter Partner**

named partner

*Partnersystem*, das mit Namen in der *Partnerliste* eingetragen ist.

**Benutzer**
user

Wird von einer *Benutzerkennung* repräsentiert. Der Begriff Benutzer ist ein Synonym für Personen, Anwendungen, Verfahren usw., die über eine Benutzerkennung Zugang zum Betriebssystem erhalten können.

**Benutzerattribute**
user attributes

Alle Merkmale einer *Benutzerkennung* in einem BS2000-System, die im *Benutzerkatalog* hinterlegt sind.

**Benutzerkatalog / Benutzerkennungskatalog**
joinfile / user catalog / user ID catalog

Datei in einem BS2000-System, die die *Benutzerattribute* aller *Benutzerkennungen* eines *Pubsets* enthält.

**Benutzerkennung / User Identification / User-Id**
user identification / user ID

In BS2000-Systemen: Maximal acht Zeichen langer Name und wird im Benutzerkatalog eingetragen. Anhand der Benutzerkennung erfolgt die Identifizierung beim Systemzugang. Alle Dateien und Jobvariablen werden unter einer Benutzerkennung eingerichtet. Die Namen der Dateien und Jobvariablen werden mit der Benutzerkennung im *Dateikatalog* hinterlegt.

**Benutzerkommando**
user command

Kommando in einem BS2000-System, das unter einer beliebigen *Benutzerkennung* im Systemmodus (/) oder auch im Programm-Modus mit CMD-Makros gegeben werden kann.

**Benutzerrechte**
user privileges

Alle an eine *Benutzerkennung* in einem BS2000-System vergebenen und im *Benutzerkatalog* hinterlegten Attribute, die Rechte darstellen.

**Benutzerverwaltung**
user administration

Siehe *Systemglobale Benutzerverwaltung*

**Berechtigungsprofil**
admission profile

Mittel zur Festlegung der Schutzfunktionen von *FTAC*. Berechtigungsprofile definieren eine *Zugangsberechtigung*, die in *FT-Aufträgen* statt der *LOGIN*- oder *LOGON-Berechtigung* angegeben werden muss. Im Berechtigungsprofil werden die *Zugriffsrechte* auf eine Benutzerkennung festgelegt, indem die Verwendung von Parametern in *FT-Aufträgen* eingeschränkt wird.

**Berechtigungsprofil, privilegiertes**
admission profile, privileged

Siehe *privilegiertes Berechtigungsprofil*

**Berechtigungssatz**
admission set

Im Berechtigungssatz wird bei Einsatz von *FTAC* für eine Benutzerkennung festgelegt mit welchen *Partnersystemen* diese Kennung welche FT-Funktionen nutzen darf.

**Berechtigungssatz, privilegierter**
admission set, privileged

Siehe *privilegierter Berechtigungssatz*

**Betriebsmittel**
resources

Hardware- und Software-Objekte, die das *FT-System* zur Ausführung eines *FT-Auftrags* benötigt (*Tasks*, Prozesse, Verbindungen, Leitungen). Diese Betriebsmittel werden durch die *Betriebsparameter* gesteuert.

**Betriebsparameter**
operating parameters

Parameter, die *Betriebsmittel* steuern (z.B. mögliche Anzahl von Verbindungen).

**Beweissicherung**
audit

Grundfunktion eines sicheren Systems; Protokollierung von Abläufen und Aufbereitung der protokollierten Daten.

**Bibliothek**
library

Datei mit interner Struktur (Elemente)

**Bibliothekselement**
library member

Teil einer Bibliothek. Ein Bibliothekselement kann seinerseits wieder in Sätze strukturiert sein.

**character repertoire**
character repertoire

Zeichenvorrat einer Datei im *virtuellen Dateispeicher*.
Für Dateien, die mit *FTAM-Partnern* übertragen werden, kann man wählen zwischen: *GeneralString*, *GraphicString*, *IA5String* und *VisibleString*.

**Character Separated Values (CSV)**

Dieses Ausgabeformat ist ein speziell im PC Umfeld weit verbreitetes, tabellenartiges Format, bei dem die einzelnen Felder durch ein Separatorenzeichen getrennt sind (häufig Semikolon ";"). Es erlaubt die Weiterverarbeitung der Ausgaben für die wichtigsten openFT-Kommandos mit eigenen Tools.

**Client**
client

- Begriff aus der Client/Server-Architektur: derjenige Partner, der die Dienste eines *Servers* in Anspruch nimmt.
- Logische Instanz, welche einem *Server* Aufträge erteilt.

**Cluster**

Eine Anzahl von Rechnern, die über ein schnelles Netzwerk verbunden sind und die von außen in vielen Fällen als ein Rechner gesehen werden können. Ziel des „Clustering" besteht meistens in der Erhöhung der Rechenkapazität oder der Verfügbarkeit gegenüber einem einzelnen Rechner.

**Comma Separated Values**

siehe *Character Separated Values*.

**Codierungsmodus**
encoding mode

Modus für die Codierung von Dateinamen, Vor-, Nach- und Folgeverbeitungen sowie ferne Kommandos, siehe *Transparenter Modus* und *Zeichenmodus*.

**Communication Controller**

Datenkommunikationsrechner

**concurrency control**

Element des FTAM-Dateiattributs *access control* (gehört zur *security group*) im *virtuellen Dateispeicher.* Steuert konkurrierende Zugriffe. Wird von openFT (BS2000) nur passiv und partiell unterstützt. Anmerkung: "partiell unterstützt" (partial support) ist ein Fachbegriff aus der FTAM-Welt, der besagt, dass der Parameter zwar syntaktisch verstanden, aber eigentlich nicht unterstützt wird.

**constraint set**

Element des FTAM-Dateiattributs *document type*.

**contents type**

FTAM-spezifisches Dateiattribut im *virtuellen Dateispeicher,* gehört zur *kernel group*, beschreibt die Dateistruktur und die Form des Dateiinhalts.

**Cross-Domain-Kopplung**

cross domain connection

Kopplung zwischen Rechnern, die sich in unterschiedlichen SNA-Domänen befinden.

Bei einer Cross-Domain-Kopplung von einem TRANSDATA-Netz (als eine SNA-Domäne) an ein SNA-Netz wird das Softwareprodukt TRANSIT-CD als *Gateway* benötigt.

**Cross-Network-Kopplung**

cross network connection

Kopplung zwischen Rechnern, die sich in unterschiedlichen SNA-Netzen befinden.

Bei einer Cross-Network-Kopplung von einem TRANSDATA-Netz an ein oder mehrere SNA-Netze wird das Softwareprodukt TRANSIT-CD sowie abhängig von der Konfiguration zusätzlich TRANSIT-SNI als *Gateway* benötigt.

**DASD (Direct Access Storage Device)**

Plattenspeicher in BS2000-Systemen.

**Data Encryption Standard (DES)**

Internationale Norm zur Verschlüsselung von Daten zur Erhöhung der Sicherheit. Das DES-Verfahren wird von den openFT-Produkten zur Verschlüsselung der Auftragsbeschreibungsdaten und ggf. der Auftragsdaten verwendet, falls mit älteren openFT-Versionen gekoppelt wird, die noch kein *AES* unterstützen.

**Dataset**

data set

Datei in z/OS.

**Dateiattribute**
file attributes
Eigenschaften einer Datei, beispielsweise Größe der Datei, Zugriffsrechte auf die Datei oder Satzstruktur der Datei.

**Dateikatalog**
file directory, file catalog
Datei in einem BS2000-System, die auf jedem *Pubset* vorhanden ist (in SM-Pubsets auf jedem Volumeset). Alle Dateien und alle Jobvariablen eines Pubsets sind im entsprechenden *Dateikatalog* eingetragen. Dateien von Privatplatten und Bändern können im Dateikatalog eingetragen sein.
Ein Katalogeintrag enthält alle Attribute (Schutzattribute, Lage der verwalteten Daten usw.) einer Datei bzw. einer Jobvariablen.

**Dateimanagement**
file management
Möglichkeit im fernen System Dateien zu „managen“. Es gibt folgende Möglichkeiten:
- Dateiverzeichnisse anlegen
- Dateiverzeichnisse anzeigen und ändern
- Dateiverzeichnisse löschen
- Dateiattribute anzeigen und ändern
- Dateien umbenennen
- Dateien löschen

**Dateispeicher, virtueller**
filestore, virtual
Siehe *virtueller Dateispeicher*

**Dateiübertragungsauftrag**
file transfer request
Siehe *FT-Auftrag*

**Dateiverzeichnis**
directory
Dateiverzeichnisse sind Ordner im hierarchischen Dateisystem eines Unix-Systems (einschließlich POSIX) oder eines Windows-Systems, welche Dateien und/oder andere Dateiverzeichnisse enthalten.
Im BS2000 (DVS) werden PLAM-Bibliotheken als Verzeichnisse interpretiert. openFT (z/OS) interpretiert einerseits den Inhalt eines PO- oder PDSE-Datasets (also die darin enthaltenen Members) als Verzeichnis, andererseits alle Dateien mit einem gemeinsamen Namensbeginn bis zu einem Qualifikationsbegrenzer (Punkt).

**Datencodierung**
data encoding

Art und Weise, in der ein *FT-System* die Zeichen intern darstellt.

**Datenkommunikationssystem**
data communication system

Summe der Hardware- und Software-Einrichtungen, die es zwei oder mehreren Kommunikationspartnern ermöglicht, unter Beachtung bestimmter Regeln Daten auszutauschen.

**Datenkomprimierung**
data compression

Reduktion von Daten durch eine verdichtete Darstellung.

**Datenschutz**
data protection

- Im engeren Sinne gemäß Bundesdatenschutzgesetz die Aufgabe, durch den Schutz der personenbezogenen Daten vor Missbrauch bei der Datenverarbeitung der Beeinträchtigung schutzwürdiger Belange der Betroffenen entgegenzuwirken.
- Im weiteren Sinne die Aufgabe, durch den Schutz der Daten vor Missbrauch in ihren Verarbeitungsphasen der Beeinträchtigung fremder und eigener schutzwürdiger Belange zu begegnen.

**Datensicherheit**
data security

Technisch-organisatorische Aufgabe, die Sicherheit von Datenbeständen und Datenverarbeitungsabläufen zu gewährleisten; d.h. insbesondere zu erreichen, dass

- der Zugriff zu Daten nur Berechtigten möglich ist,
- keine unerwünschte bzw. unberechtigte Verarbeitung von Daten erfolgt,
- die Daten bei der Verarbeitung nicht verfälscht werden,
- die Daten reproduzierbar sind.

**DHCP**

Dienst in TCP/IP-Netzen, der Clients auf Anforderung automatisch IP-Adressen und TCP/IP-Parameter zuteilt.

**Dienst**
service

- Begriff der OSI-Architektur: Ein Dienst (Service) ist die Menge von Funktionen, die ein Diensterbringer (Service Provider) an einem Dienstzugangspunkt (Service Access Point) zur Verfügung stellt.

- Begriff der Client-Server-Architektur: Eine Menge von Funktionen, die ein Server den Clients zur Verfügung stellt.
- Begriff in Unix- und Windows-Systemen: Ein Programm, eine Routine oder ein Prozess zur Durchführung einer bestimmten Systemfunktion, die der Unterstützung anderer Programme dient, insbesondere auf einer niedrigen (hardwarenahen) Ebene.

**Direct Access Storage Device (DASD)**

Plattenspeicher in BS2000-Systemen.

**document type**

Wert des FTAM-spezifischen Dateiattributs *contents type* (gehört zur *kernel group*). Beschreibt die Form des Dateiinhalts im *virtuellen Dateispeicher*.

- *document type* für Textdateien: **FTAM-1**
- *document type* für Binärdateien: **FTAM-3**

**Dynamischer Partner**

dynamic partner

*Partnersystem*, das entweder gar nicht (*freier dynamischer Partner*) oder nur mit Adresse und ohne Namen (*eingetragener dynamischer Partner*) in der Partnerliste eingetragen ist.

**EBCDIC**

Normierter Standardcode für den Nachrichtenaustausch, wie er z.B. in BS2000- oder z/OS-Systemen vorkommt. Das Akronym steht für „Extended Binary Coded Decimal Interchange Code“.

**Eigentümer eines FT-Auftrags**

owner of an FT request

Benutzerkennung im *lokalen System* bzw. *fernen System*, mit der dieser *FT-Auftrag* durchgeführt wird. Eigentümer ist immer die Kennung, unter der der Auftrag abgesetzt wurde, nicht diejenige unter der der Auftrag durchgeführt wird.

**Eingetragener dynamischer Partner**

registered dynamic partner

Partnersystem, das nur mit Adresse und ohne Namen in der Partnerliste eingetragen ist.

**Empfangsdatei**

receive file

Datei im *Empfangssystem*, in der die Daten einer *Sendedatei* abgespeichert werden.

**Empfangssystem**
receive system
: System, an das eine Datei gesendet wird. Dies kann das *lokale* oder *ferne System* sein.

**Emulation**
emulation
: Komponente, die die Eigenschaften eines anderen Geräts nachbildet.

**Entity**
entity
: Siehe *Instanz*

**Ergebnisliste**
result list
: In BS2000- und z/OS-Systemen: Liste mit Informationen über eine beendete Dateiübertragung, die der Benutzer im *lokalen System* für von ihm gestellte *FT-Aufträge* erhält.

**Explorer**
: Programm von Microsoft, das zusammen mit Windows-Betriebssystemen ausgeliefert wird und eine einfache Art der Navigation im Dateisystem ermöglicht.

**Fernadministration**
remote administration
: Administration von openFT-Instanzen von fernen Rechnern aus.

**Fernadministrations-Server**
remote administration server
: Zentrale Komponente, die für die *Fernadministration* und für *ADM-Traps* benötigt wird. Ein Fernadministrations-Server läuft auf einem Unix- oder Windows-System mit openFT ab V11.0. Wenn er für die *Fernadministration* eingesetzt wird, dann enthält er sämtliche dafür notwendigen Konfigurationsdaten.

**Fernadministrator**
remote administrator
: Rolle, die im *Fernadministrations-Server* konfiguriert wird und dazu berechtigt, bestimmte Administrationsfunktionen auf bestimmten openFT-Instanzen auszuführen.

**Fernes System**
remote system
: Siehe *Partnersystem*

**File Management**
file management
Siehe *Dateimanagement*

**File Transfer**
file transfer
Dateiübertragung

**Firewall-Rechner**
firewall processor
Rechner, der zwei Netze miteinander verbindet. Die möglichen Zugriffe können genau geregelt und auch protokolliert werden.

**Folgeverarbeitung**
follow-up processing
FT-Funktion, die nach Abschluss des *FT-Auftrages* die vom Benutzer spezifizierten Kommandos oder Anweisungen im *lokalen* und/oder *fernen System* zur Ausführung bringt. Für positiven und negativen Abschluss können unterschiedliche Folgeverarbeitungen definiert werden, siehe auch *Vor- und Nachverarbeitung*.

**Folgeverarbeitungsauftrag**
follow-up processing request
Anweisungen innerhalb eines *FT-Auftrages*, die nach der Dateiübertragung *Folgeverarbeitung(en)* durchführen.

**Freier dynamischer Partner**
free dynamic partner
Partnersystem, das nicht in der Partnerliste eingetragen ist.

**FTAC (File Transfer Access Control)**
Erweiterter Zugangsschutz bei Dateiübertragung und Dateimanagement. Für BS2000 und z/OS realisiert im Produkt openFT-AC, für andere Betriebssysteme Bestandteil des openFT-Produkts, z.B. bei openFT (Unix-Systeme) und openFT (Windows).

**FTAC-Logging-Funktion**
FTAC logging function
Funktion, mit der FTAC jeden Zugriff über File Transfer auf das geschützte System protokolliert.

**FTAC-Verwalter**
FTAC administrator

Verwalter der FTAC-Funktionen, sollte mit demjenigen identisch sein, der für den Datenschutz verantwortlich ist.
Der FTAC-Verwalter legt für sein System u.a. die sicherheitstechnischen Rahmenbedingungen in Form eines für alle geltenden Standardberechtigungssatzes fest.

**FTAC-Zugangsberechtigung**
FTAC transfer admission

Berechtigung für die Dateiübertragung und das Dateimanagement bei Einsatz von FTAC. Die Zugangsberechtigung ersetzt die *LOGIN-Berechtigung* bzw. die *LOGON-Berechtigung*.

**FTADM-Protokoll**
FTADM protocol

Protokoll, das bei der Kommunikation zwischen zwei openFT-Instanzen verwendet wird, um *Fernadministration* zu betreiben oder *ADM-Traps* zu übertragen.

**FTAM-1**

*document type* für Textdateien

**FTAM-3**

*document type* für Binärdateien

**FTAM-Dateiattribute**
FTAM file attributes

Jedes System, das den File Transfer über FTAM-Protokolle ermöglicht, muss seine Dateien dem Partner in einer normgemäßen Beschreibung (ISO 8571) zur Verfügung stellen. Zu diesem Zweck werden die Attribute einer Datei vom realen Dateispeicher auf einen *virtuellen Dateispeicher* abgebildet und umgekehrt. Dazu werden im Wesentlichen drei Gruppen von Dateiattributen unterschieden:

– kernel group: beschreibt die wesentlichen Attribute der Dateien.
– storage group: umfasst die Speicherattribute von Dateien.
– security group: definiert Sicherheitsattribute bzgl. Zugang und Zugriff.

**FTAM-Katalog**
FTAM catalog

Der FTAM-Katalog dient auf Unix- und Windows-Systemen zur Erweiterung der verfügbaren Dateiattribute. Die Erweiterung ist nur bei Zugriffen über FTAM von Bedeutung. Zum Beispiel kann eine Datei auf einem Unix-System mit dem Kommando *rm* und auf einem Windows-System mit dem Kommando *erase* gelöscht werden, auch wenn der Parameter *permitted actions* das nicht erlaubt.

**FTAM-Partner**
FTAM partner

*Partnersystem*, mit dem über *FTAM-Protokolle* kommuniziert wird.

**FTAM-Protokoll (File Transfer, Access and Management)**
FTAM protocol

Von der ISO (International Organization for Standardization) genormtes *Protokoll* für die Dateiübertragung (ISO 8571, FTAM).

**FTP-Partner**
FTP partner

Partnersystem, mit dem über das *FTP-Protokoll* kommuniziert wird.

**FTP-Protokoll**
FTP protocol

Herstellerunabhängiges Protokoll zur Dateiübertragung in TCP/IP-Netzen.

**FT-Auftrag**
FT request

Auftrag an ein *FT-System*, eine Datei von einem *Sendesystem* zu einem *Empfangssystem* zu übertragen und gegebenenfalls *Folgeverarbeitungsaufträge* zu starten.

**FT-Profil**
FT profile

siehe *Berechtigungsprofil*.

**FT-System**
FT system

System zur Dateiübertragung, bestehend aus einem Rechner und der zur Dateiübertragung nötigen Software.

**FT-Trace**
FT trace

Diagnosefunktion, die den Ablauf des FT-Betriebs protokolliert.

**FT-Verwalter**
FT administrator

Person, die das openFT-Produkt auf einem Rechner verwaltet, d.h. u.a. für die Einträge in die *Partnerliste* und die Steuerung der Betriebsmittel zuständig ist. Auf Unix-Systemen kann openFT von allen Benutzerkennungen mit UID=0 verwaltet werden.

**Funktionsnorm**
functional standard

Empfehlung, wann und wie bestimmte ISO-/OSI-Normen eingesetzt werden sollen (äquivalenter Begriff: *Profil*). Für die Übertragung unstrukturierter Dateien ist die europäische Vornorm CEN/CENELEC ENV 41 204 erstellt worden, für das Dateimanagement die europäische Vornorm CEN/CENELEC ENV 41205.

**Gateway**
gateway

Im allgemeinen Sprachgebrauch ein System, das zwei oder mehr Netze miteinander verknüpft und nicht als Bridge arbeitet. Varianten: Gateway auf Netzebene (d.h. Router oder OSI-Relais), Transport- und Anwendungsgateway.

**Gateway-Rechner**
gateway processor

*Kommunikationsrechner*, die ein Rechnernetz mit einem anderen Rechnernetz verbinden. In Gateway-Rechnern werden die unterschiedlichen Protokolle der unterschiedlichen Rechnernetze aufeinander abgebildet.

**Gemeinschaftlicher Datenspeicherbereich**
public space

Benannter Plattenspeicherbereich in BS2000-Systemen, der für eine definierte Anzahl von Benutzerkennungen des Betriebssystems verfügbar ist. Dieser Speicherbereich kann sich über einen oder mehrere Public Volume Sets (*Pubsets*) erstrecken.

**Generalized Trace Facility (GTF)**

IBM-Tool zur Erstellung von Traces (insbesondere für die Überwachung des Datenverkehrs zwischen einem Anwendungsprogramm und den zugehörigen VTAM-Applikationen sowie zwischen VTAM-Applikationen und der DFÜ-Leitung).

**GeneralString**

Zeichenvorrat für Dateien, die mit *FTAM-Partnern* übertragen werden

**GraphicString**
: Zeichenvorrat für Dateien, die mit *FTAM-Partnern* übertragen werden

**Globale Auftrags-Identifikation**
global request identification / global request ID / global request number
: Auftragsnummer, die der *Initiator* bei einem openFT- oder FTAM-Auftrag an den *Responder* übermittelt. D.h. die globale Auftrags-Identifikation im Responder ist identisch zur *Auftrags-Identifikation* im Initiator. Der Responder erzeugt für den Auftrag eine eigene (lokale) Auftrags-Identifikation. Damit lassen sich vor allem für den Wiederanlauf-Fall die im Initiator und im Responder gespeicherten Informationen eindeutig einem Auftrag zuordnen.

**Grundfunktionen**
basic functions
: Die wichtigsten File-Transfer-Funktionen. Im *Berechtigungssatz* wird die Menge der Grundfunktionen festgelegt, die von einer Benutzerkennung genutzt werden können. Die sechs Grundfunktionen sind:
  – Inbound Empfangen
  – Inbound Senden
  – Inbound Folgeverarbeitung
  – Inbound Dateimanagement
  – Outbound Empfangen
  – Outbound Senden

**Guard**
: Objekt der Bedingungsverwaltung GUARDS in BS2000-Systemen. In einem Guard werden Bedingungen gesammelt, die von der Standard-Bedingungsverwaltung von GUARDS auf Anfrage ausgewertet werden.

**GUARDS (Generally Usable Access Control Administration System)**
: Objektverwaltung für *Guards*.

**heterogenes Netz**
heterogeneous network
: Ein Netz, das aus mehreren Teilnetzen aufgebaut ist, die nach unterschiedlichen technischen Prinzipien arbeiten.

**Hintergrundprozess**
background process
: Ein Prozess in Unix-Systemen, der unabhängig vom Benutzerprozess abläuft. Man erzeugt einen Hintergrundprozess, indem man das Kommando mit dem Sonderzeichen & abschließt. Anschließend ist der Prozess, der den Hinter-

grundprozess abschickt, sofort für neue Aufgaben frei und braucht sich um den Hintergrundprozess nicht mehr zu kümmern, der nun simultan zu ihm selbst abläuft.

**homogenes Netz**
homogeneous network
Ein technisch nach einem einzigen Prinzip aufgebautes Netz.

**Host**
host
Früher ein großes Datenverarbeitungssystem, das zur Kommunikation einen *Front End Processor* benötigte. Heute Bezeichnung für BS2000- oder z/OS-Systeme.

**HOSTS-Datei**
HOSTS file
Netzverwaltungsdatei in Unix- und Windows-Systemen, die alle erreichbaren Rechner mit Internet-Adresse, Rechnername und Alias-Name enthält.

**IA5String**
Zeichenvorrat für Dateien, die mit *FTAM-Partnern* übertragen werden.

**Identifizierung**
identification
Verfahren zur Erkennung einer Person oder eines Objekts.

**IEBCOPY**
IBM-Tool für das Kopieren von Bibliotheken (PO- oder PDSE-Datasets).

**IEBGENER**
IBM-Tool für das Kopieren von sequentiellen Dateien (PS-Datasets).

**IEBPTPCH**
IBM-Tool für das Drucken von Dateien.

**inbound-Auftrag**
inbound request / inbound submission
Auftrag, der in einem anderen System gestellt wurde.

**Inbound Dateimanagement**
inbound file management
In einem *fernen System* gestellter *Auftrag*, bei dem Dateiverzeichnisse oder Dateiattribute des *lokalen Systems* angesehen, Dateiattribute geändert sowie lokale Dateien gelöscht werden können.

**Inbound Empfangen**
inbound receive
: In *fernem System* gestellter *Auftrag*, bei dem im *lokalen System* eine Datei empfangen wird.

**Inbound Folgeverarbeitung**
inbound follow-up processing
: In einem *fernen System* gestellter *Auftrag* mit *Folgeverarbeitung* im *lokalen System*.

**Inbound Senden**
inbound send
: In *fernem System* gestellter *Auftrag*, bei dem eine Datei aus dem *lokalen System* in das ferne System gesendet wird.

**Initiator**
initiator
: Hier: *FT-System*, in dem ein *FT-Auftrag* gestellt wurde.

**Instanz / Entity**
instance / entity
: Begriff der OSI-Architektur: aktives Element in einer Schicht. Siehe auch *openFT-Instanz*

**Instanzidentifikation**
instance ID
: Netzweit eindeutige Adresse einer *openFT-Instanz*.

**Integrität**
integrity
: Unverfälschtheit und Korrektheit von verarbeiteten, übertragenen und gespeicherten Daten.

**Interactive Problem Control System (IPCS)**
: IBM-Tool zur Aufbereitung eines maschinenlesbaren (unformatierten) Dumps.

**interoperability**
interoperability
: Fähigkeit zweier *FT-Systeme*, zusammenarbeiten zu können.

**ISO-/OSI-Referenzmodell**
ISO/OSI reference model
: Das ISO-/OSI-Referenzmodell stellt einen Rahmen für die Normung der Kommunikation offener Systeme dar (ISO=International Organization for Standardization).

**ISPF, ISPF/PDF**

Menügeführte Dienste in z/OS für die Entwicklung von Software und für die Führung eines (TSO-) Dialogs.

**Job**

Folge von Kommandos, Anweisungen und Daten.
Auf z/OS Folge von JCL-Anweisungen (Batch).

**Jobklasse**
job class

In Jobklassen werden *Jobs* in BS2000-Systemen zusammengefasst, die in bestimmten Eigenschaften und Merkmalen übereinstimmen.

**Jobübertragung**
job transfer

Übertragung einer Datei, die im *Empfangssystem* einen *Job* darstellt und dort als solcher angestoßen wird.

**Kennwort / Passwort**
password

Folge von Zeichen, die der Benutzer eingeben muss, um den Zugriff zu einer Benutzerkennung, einer Datei, einer Jobvariablen, einem Netzknoten oder einer Anwendung zu erhalten. Das Kennwort einer Benutzerkennung dient zur *Authentifizierung* des Benutzers. Es dient dem Zugangsschutz. Das Datei-Kennwort dient zur Überprüfung der Zugriffsberechtigung beim Zugriff auf eine Datei (Jobvariable).

**Kernel group**

Gruppe von FTAM-spezifischen Dateiattributen im *virtuellen Dateispeicher*, umfasst die Kernattribute einer Datei.

**Kommunikationsrechner**
communication computer

Rechner zum Aufbau eines *Datenkommunikationssystems*.

**Komprimierung**
compression

Bedeutet, dass mehrere aufeinanderfolgende identische Zeichen auf ein Zeichen verkürzt werden und die Zeichenanzahl hinzugefügt wird. Damit verkürzen sich Übertragungszeiten.

**Konfigurations-Userid**
configuration user ID

Jede openFT-Instanz in einem BS2000-System benötigt eine Kennung, auf der die variablen Dateien dieser openFT-Instanz hinterlegt sind (für die Standardinstanz: $SYSFJAM).

**Konnektivität**
connectivity

Allgemein die Möglichkeit der Kommunikation zwischen Systemen und Partnern, gelegentlich auch nur auf die Kommunikationsmöglichkeiten von Transportsystemen bezogen.

**Local Area Network (LAN)**

Ursprünglich ein mit hoher Geschwindigkeit arbeitendes Netz geringer Reichweite. Heute jedes Netz auch großer Reichweite, das gemäß CSMA/CD, Token Ring oder FDDI arbeitet (vgl. *WAN Wide Area Network*).

**Logging-Funktion**
logging function

Funktion, mit der *openFT* jeden Zugriff über File Transfer auf das geschützte System protokolliert.

**Logging-Satz**
logging record

Enthält Informationen über eine von openFT durchgeführte Zugangsprüfung (FTAC-Logging-Satz) oder über einen Übertragungs- oder Fernadministrations-Auftrag, der nach positiver Zugangsprüfung gestartet wurde (FT-Logging-Satz bzw. ADM-Logging-Satz).

**Logical Unit (LU)**

Schnittstelle zwischen einem Anwendungsprogramm und dem Datenkommunikationsnetz SNA. Der Typ einer LU beschreibt die Charakteristika der Kommunikation.

**Login-Berechtigung**
login authorization

Berechtigung für den Zugriff auf einen Rechner. Die Login-Berechtigung, die (in der Regel) aus Benutzerkennung und Kennwort besteht, berechtigt zum Dialogbetrieb, siehe auch *LOGON-Berechtigung*.

**LOGON-Berechtigung**
LOGON authorization
: Berechtigung für den Zugriff auf einen Rechner. Die LOGON-Berechtigung, die (in der Regel) aus Benutzerkennung, Abrechnungsnummer und Kennwort besteht, berechtigt zum Dialogbetrieb.

**Lokales System**
local system
: Das *FT-System*, an dem der Benutzer arbeitet.

**maximum-string-length**
maximum-string-length
: Bezeichnet im *virtuellen FTAM-Dateispeicher* die maximale Länge von *Strings* (Zeichenketten) innerhalb einer Datei.

**Nachverarbeitung**
postprocessing
: openFT bietet die Möglichkeit, im Empfangssystem die empfangenen Daten durch eine Folge von Betriebssystemkommandos bearbeiten zu lassen. Die Nachverarbeitung läuft dabei (im Gegensatz zur *Folgeverarbeitung*) unter der Prozesskontrolle von openFT.

**Network Control Program (NCP)**
: Betriebssystem des Front-End-Prozessors für einen SNA-Host.

**NEA**
: Bezeichnung einer Netzarchitektur in BS2000-Systemen.

**NetMaster**
: Tool für die Steuerung eines Datenkommunikationssystems.

**NetView**
: IBM-Tool für die Steuerung eines Datenkommunikationssystems.

**Network Management Kernel**
: Bestandteil der Network Management Plattform in BS2000-Systemen; zuständig für die Vermittlung von Netzmanagement-Aufträgen sowie für zentrale Aufgaben wie Logging, Berechtigungsprüfung, Auftrags- und Anwendungsverwaltung.

**Netzbeschreibungsbuch**
network description file
: Datei, die bis openFT V9 verwendet wurde und die Angaben über *ferne Systeme* (*FT-Systeme*) enthält.

**Offenes Rechnernetz**
open computer network

Rechnernetz, in dem nach den Regeln von ISO/OSI kommuniziert wird. Durch festgelegte *Protokolle* wird das Zusammenwirken unterschiedlicher Rechner verschiedener Hersteller möglich.

**Offline Logging**
offline logging

Die Logging-Datei kann im laufenden Betrieb gewechselt werden. Nach dem Umschalten bleibt die bisherige Logging-Datei als Offline-Logging-Datei bestehen; neue Logging-Sätze werden in eine neue Logging-Datei geschrieben. Die Logging-Sätze einer Offline-Logging-Datei können weiterhin mit openFT-Mitteln angesehen werden.

**openFT Explorer**
openFT Explorer

Programm von openFT (Unix-Systeme) und openFT (Windows), das eine grafische Oberfläche zur Verfügung stellt, über die Dateiübertragungs- und Administrations-Funktionen von openFT durchgeführt werden können.

**openFT Monitor**
openFT Monitor

Programm von (Unix-Systeme) und openFT (Windows), mit dem die Messdaten des openFT-Betriebs in Diagrammform angezeigt werden können. Der openFT Monitor benötigt ein grafikfähiges Terminal.

**openFT-FTAM**

Zusatzprodukt zu openFT (für BS2000, Unix-Systeme und Windows-Systeme) zur Unterstützung der Dateiübertragung mit FTAM-Protokollen. FTAM steht für File Transfer, Access and Management (ISO 8571).

**openFT-Installationsverzeichnis**
openFT installation directory

Pfad, unter dem openFT (Windows) installiert ist. Der Pfad kann bei einer interaktiven Installation frei gewählt werden. Bei der bedienerlosen Installation kann er im Parameter INSTALLDIR festgelegt werden. Der Standard-Pfad ist abhängig von der Spracheinstellung und der Version des Windows-Betriebssystems
(Standardwert: %Program Files%\openFT).

**openFT-Instanz**
openFT instance
Auf einem einzelnen Rechner oder einem Cluster im HIPLEX-Verbund, TCP/IP-Netz oder Sysplex-Verbund können mehrere openFT-Systeme gleichzeitig ablaufen, sogenannte openFT-Instanzen. Jede Instanz hat eine eigene Adresse (Instanzidentifikation, virtueller BCAM-Host in BS2000-Systemen, Host in z/OS-Systemen) und besteht aus dem geladenen Code der openFT-Produkte (inklusive der Zusatzprodukte sofern vorhanden) und den variablen Dateien wie *Partnerliste*, Logging-Dateien, Schlüsselbibliothek, Auftragsbuch usw.

**openFT-Partner**
openFT partner
*Partnersystem*, mit dem über *openFT-Protokolle* kommuniziert wird.

**openFT-Protokolle**
openFT protocols
Genormte *Protokolle* für die Dateiübertragung (SN77309, SN77312).

**openFT-Script**
openFT-Schnittstelle für Unix- und Windows-Systeme, die eine XML-basierte Skript-Sprache für Dateiübertragungs- und Dateimanagement-Funktionen zur Verfügung stellt. Diese Schnittstelle ermöglicht es, mehrere Dateiübertragungs- oder Dateimanagementaufträge zu einem openFT-Script-Auftrag zusammen zu fassen.

**openFT-Script-Kommandos**
openFT-Script commands
Kommandos zur Verwaltung von openFT-Script-Aufträgen in Unix- und Windows-Systemen.

**Outbound-Auftrag**
outbound request / outbound submission
Auftrag, der im eigenen Rechner gestellt wurde.

**Outbound Empfangen**
outbound receive
Lokal gestellter Auftrag bei dem im *lokalen System* eine Datei empfangen wird.

**Outbound Senden**
outbound send
Lokal gestellter Auftrag, bei dem aus dem *lokalen System* eine Datei gesendet wird.

**Owner**

Siehe *Eigentümer eines FT-Auftrags*

**Partitioned Data Set Extended (PDSE-Dataset)**

partitioned data set extended

Bibliothek des IBM-z/OS-Datenverwaltungssystems. Enthält einzelne Elemente (Members). Kann anstelle eines Partitioned Organized Data Sets verwendet werden. Voraussetzung für die Nutzung von PDSE ist das IBM-Software-Produkt "Data Facility Storage Management Subsystem" (DFSMS).

**Partitioned Organized Dataset (PO-Dataset)**

partitioned organized data set

Bibliothek des IBM-z/OS-Datenverwaltungssystems. Enthält einzelne Elemente (Members).

**Partner**

partner

siehe *Partnersystem*

**Partnerliste**

partner list

Datei, die Angaben über *ferne Systeme* (*FT-Systeme*) enthält.

**Partnersystem**

partner system

Hier: *FT-System*, das zusammen mit dem *lokalen System FT-Aufträge* durchführt.

**permitted actions**

FTAM-spezifisches Dateiattribut im *virtuellen Dateispeicher,* gehört zur *kernel group*, legt grundsätzlich erlaubte Aktionen fest.

**Personenbezogene Beweissicherung**

Personal Audit for Individual Accountability

Nachvollziehbarkeit des Umgangs mit einem System. Identifikation entweder in Form:

- eine Benutzerkennung entspricht einem Benutzer oder
- ein Benutzer darf ausschließlich eine Bedienstation benutzen.

**Physical-sequential Dataset (PS-Dataset)**

physical sequential data set

Sequentiell organisierte Datei des IBM-z/OS-Datenverwaltungssystems, entspricht ungefähr einer BS2000-SAM-Datei.

**Physical Unit (PU)**

Jeder Knoten eines SNA-Netzes enthält als adressierbare Instanz eine Physical Unit (PU). Sie ist zuständig für die Verbindungsüberwachung zum Host und für die Überwachung der *Logical Units* (LUs).

**Portnummer**

port number

Nummer, die eine TCP/IP-Anwendung bzw. den Endpunkt einer TCP/IP-Verbindung innerhalb eines Rechners eindeutig identifiziert.

**POSIX (Portable Open System Interface)**

Gremium und von ihm geschaffene Normen für auf verschiedene Systemplattformen portable Schnittstellen.

**Postkorb**

mailbox

Der Postkorb ist eine Datei in einem Unix-System, die mit dem Kommando mail gelesen werden kann. Jeder Benutzer hat einen Postkorb, um darin Nachrichten zu empfangen.

**Presentation**

presentation

Instanz zur Realisierung der Darstellungsschicht (Schicht 6) des *ISO/OSI-Referenzmodells* in einem *FT-System*, das z.B. mit *FTAM-Protokollen* arbeitet.

**Presentation-Selektor**

presentation selector

Subadresse, mit der eine *Presentation-Anwendung* angesprochen wird.

**Private key**

Geheimer Dechiffrierschlüssel, mit dem der Empfänger eine mit einem *public key* verschlüsselte Nachricht entschlüsseln kann. Wird von verschiedenen Verschlüsselungsverfahren verwendet, u.a. dem *RSA-Verfahren*.

**Privileg**

privilege

- In BS2000 und z/OS-Systemen: Systemglobales Recht, das zur Ausführung bestimmter Kommandos und zum Aufrufen bestimmter Programmschnittstellen berechtigt (z.B. TSOS-Privileg).
- In BS2000-Systemen: Satz von Attributen eines Benutzers, die vom Zugriffskontrollsystem benutzt werden

**privilegiertes Berechtigungsprofil**
privileged admission profile

*Berechtigungsprofil*, mit dem ein Benutzer die Vorgaben des *FTAC-Verwalters* im *Berechtigungssatz* überschreiten kann. Dazu bedarf es der Genehmigung des *FTAC-Verwalters*. Nur er kann Berechtigungsprofile privilegieren.

**privilegierter Berechtigungssatz**
privileged admission set

*Berechtigungssatz* des *FTAC-Verwalters*.

**Profil**
profile

Bei OSI eine Norm, die für einen bestimmten Zweck festschreibt, welche Protokolle einzusetzen sind und Vorschriften über die Werte der Parameter und Optionen enthält.
Hier: Ein einer Benutzerkennung zugeordneter Kommando-Vorrat, dessen Zulässigkeit über Syntax-Dateien sichergestellt wird.
Siehe auch *Berechtigungsprofil*, *privilegiertes Berechtigungsprofil*.

**Prompting in Prozeduren**
prompting in procedures

Funktion in Unix- und Windows-Systemen, die den Datenstationsbenutzer auffordert, zum Ablauf der Prozedur benötigte Daten einzugeben.

**Protokoll**
protocol

Summe der Regeln und Verfahren zwischen zwei oder mehr gleichrangigen Partnern, um einen festgelegten Zweck zu erreichen, meist in Form der Definition der auszutauschenden Nachrichten und der korrekten Abläufe von Nachrichtenfolgen inklusive der Behandlung von Fehlerfällen und sonstigen Ausnahmefällen.

**Prozedur**
procedure

In z/OS: Kommando-Prozedur, entspricht im Allgemeinen einer IBM CLIST oder REXX-Prozedur.

**Public key**

Veröffentlichter Chiffrierschlüssel; wird vom Empfänger einer Nachricht festgelegt und veröffentlicht bzw. dem Absender der Nachricht mitgeteilt, damit dieser an den Empfänger gerichtete Nachrichten damit verschlüsseln kann. Wird von verschiedenen Verschlüsselungsverfahren verwendet, u.a. dem Rivest-Shamir-Adleman-Verfahren (*RSA-Verfahren)*; muss zu dem nur dem Empfänger bekannten *private key* passen.

**Pubset / Public Volume Set**

*Satz* gemeinschaftlich gekennzeichneter Plattenspeicher-Einheiten in einem BS2000-System, durch eine Katalogkennung (Catid) definiert. Pubsets werden in *SF-Pubsets* und SM-Pubsets unterschieden.

**RACF**

Produkt von IBM für die Zugangs- und Zugriffssicherung.

**RAS**

Remote Access Service; ein Dienst von Windows, der die Kommunikation mit fernen Rechnern ermöglicht.

**Relais**

relay

Bezeichnung in OSI für ein Element einer Schicht, das zwischen zwei anderen Partnern vermittelt und so erst die Kommunikation zwischen diesen beiden Partnern ermöglicht.
Im engeren Sinn entspricht ein Relais auf der Netzschicht funktionell einem *Router*.

**Relaisprogramm**

relay program

Programm in einem *Gateway-Rechner*, das die unterschiedlichen Protokolle aufeinander abbildet.

**relativer Pfadname**

relative path name

Weg vom gerade aktuellen *Dateiverzeichnis* bis zur Datei.

**Responder**

responder

Hier: *FT-System*, welches vom *Initiator* angesprochen wird.

**REXX**

Prozedursprache der IBM.

**RFC (Request for Comments)**

Verfahren im Internet zur Kommentierung von vorgeschlagenen Normen, Festlegungen oder auch Berichten. Auch Bezeichnung für ein auf diese Weise verabschiedetes Dokument.

**RFC1006**

Zusatzprotokoll zur Realisierung der ISO-Transportdienste (Transportklasse 0) auf TCP/IP-Basis.

**Rivest-Shamir-Adleman-Verfahren (RSA-Verfahren)**
Rivest Shamir Adleman (RSA) procedure

Nach seinen Erfindern benanntes Verschlüsselungsverfahren, das mit einem aus *public key* und *private key* bestehenden Schlüsselpaar arbeitet. Wird von der openFT-Produktfamilie benutzt, um die Identität des Partnersystems eindeutig zu überprüfen und dem Partnersystem den AES-Schlüssel für die Verschlüsselung der Dateiinhalte zu übermitteln.

**Router**
router

Element in einem Netz, das zwischen Netzen residiert und Nachrichtenströme durch die Netze lenkt und dazu Wegewahl, Adressierung und andere Funktionen behandelt. Arbeitet auf Schicht 3 des OSI-Modells.

**RPC (Remote Procedure Call)**

Aufruf einer Prozedur im Server vom Client aus über Netze hinweg.

**Satz**
record

Eine Zusammenfassung von Daten, die als eine logische Einheit behandelt werden.

**Satz fester Länge**
fixed-length record

Ein Satz in einer Datei, in der alle Sätze nach Vereinbarung dieselbe Länge haben; innerhalb der Datei ist keine Anzeige der Länge erforderlich.

**Satz variabler Länge**
variable length record

Satz in einer Datei, in der die Sätze unterschiedlich lang sein können. Die Satzlänge muss entweder durch ein Satzlängenfeld am Anfang des Satzes angegeben werden, oder implizit durch einen Begrenzer (z.B. Carriage Return - Line Feed) zum nächsten Satz ermittelbar sein.

**Schutzattribute**
security attributes

Sicherheitsrelevante Eigenschaften eines Objekts, die Art und potenzielle Möglichkeit des Zugriffs auf dieses Objekt festlegen.

**Secure FTP**
secure FTP

Verfahren, mit dem eine Verbindung über das *FTP-Protokoll* getunnelt wird, so dass sichere Verbindungen mit Verschlüsselung und *Authentifizierung* möglich sind.

**Security group**
security group
Gruppe von FTAM-spezifischen Dateiattributen des *virtuellen Dateispeichers*, umfasst die Sicherheitsattribute einer Datei.

**Sendedatei**
send file
Datei im *Sendesystem*, aus der Daten in die *Empfangsdatei* gesendet werden.

**Sendesystem**
sending system
Hier: *FT-System*, das eine Datei sendet. Dies kann das *lokale* oder das *ferne System* sein.

**Server**
server
Logische Instanz bzw. Anwendungskomponente, welche Aufträge eines Clients ausführt und die (koordinierte) Nutzung allgemein verfügbarer Dienste (File, Print, Datenbank, Kommunikation, etc.) bereitstellt. Kann selbst bezüglich eines anderen Servers Client sein.

**Service Class**
service class
Parameter, mit dem *FTAM-Partner* aushandeln, welche Funktionalität sie verwenden.

**Session**
session
– In OSI die Bezeichnung für eine Schicht-5-Verbindung.
– In SNA eine allgemeine Bezeichnung für eine Verbindung zwischen Kommunikationspartnern (Applikationen, Geräten oder Benutzern).

**Session-Selektor**
session selector
Subadresse, mit der eine *Session-Anwendung* angesprochen wird.

**SF-Pubset (Single Feature Pubset)**
SF pubset
Ein oder mehrere Platten in einem BS2000-System, die in den wesentlichen Eigenschaften (Plattenformat, Allokierungseinheit) übereinstimmen und der Ablage von Dateien und JVs unter einer gemeinsamen Katalogkennung dienen.

**Shell-Metazeichen**
shell metacharacters

Folgende Metazeichen haben auf Unix- und Windows-Systemen eine besondere Bedeutung für die Unix-Shell bzw. Windows-Eingabeaufforderung: *, [ ], ?, <, >, |, &, &&, ( ), { }

**Sicherheitsstufe**
security level

Bei Einsatz von FTAC bzw. der *FTAC-Funktionen* ist die Sicherheitsstufe ein Maß für das Schutzbedürfnis gegenüber einem *Partnersystem*.

**SMF (Service Management Facility)**

Tool für die Steuerung von Services auf Solaris-Systemen.

**SMF (System Management Facility)**

IBM-Tool für das Sammeln von Abrechnungs- und Statistik-Daten.

**SMP/E (System Modification Program/Extended)**

IBM-Produkt zum Installieren und Verwalten von Software-Produkten, deren Versionen und Korrekturen.

**SNA-Netz**
SNA network

*Datenkommunikationssystem*, das sich entsprechend der Systems Network Architecture (SNA) von IBM verhält.

**SNMP (Simple Network Management Protocol)**

Von der Internet Engineering Task Force (IETF) für TCP/IP-Netze definiertes Protokoll zur Übertragung von Managementinformationen.

**Sonderzeichen**
special characters

Siehe *Shell-Metazeichen*

**Standardausgabe (stdout)**
standard output

Standardausgabe ist auf Unix- und Windows-Systemen voreingestellt auf den Bildschirm.

**Standardberechtigungssatz**
standard admission set

Der Standardberechtigungssatz ist die Vorgabe für alle Benutzerkennungen. Der Benutzer darf diese Vorgabe für seinen Berechtigungssatz weiter einschränken.

**Standardeingabe (stdin)**
standard input
: Standardeingabe ist auf Unix- und Windows-Systemen voreingestellt auf die Tastatur.

**Standardfehlerausgabe (stderr)**
standard error output
: Standardfehlerausgabe ist auf Unix- und Windows-Systemen voreingestellt auf den Bildschirm.

**Standardinstanz**
standard instance
: Die erste openFT-Instanz, die immer vorhanden ist und beim Start von openFT aktiviert bzw. geladen wird. An sie wenden sich standardmäßig alle openFT-Kommandos und Programmschnittstellenaufrufe, wenn keine andere Instanz eingestellt wurde.

**Standardzugriffskontrolle**
Standard Access Control
: BS2000-Systeme: Besteht aus den in den Kommandos CREATE-FILE oder MODIFY-FILE-ATTRIBUTES festgelegten Zugriffsrechten ACCESS und USER-ACCESS.

**Storage group**
storage group
: FTAM-spezifisches *Dateiattribut* im *virtuellen Dateispeicher*, umfasst die Speicherattribute von Dateien.

**String**
string
: Zeichenkette

**string-significance**
string-significance
: Beschreibt für die Übertragung mit *FTAM-Protokollen* das Format der *Strings* in den Dateien.

**Subsystem**
subsystem
: In BS2000-Systemen: Teil eines Systems, der einen abgeschlossenen Funktionskomplex bearbeitet.

**SU-Privileg**
SU privilege
: Privileg eines FTAC-Verwalters im z/OS, das ihn berechtigt, auf fremden Benutzerkennungen ohne Kenntnis des aktuellen Benutzerkennwortes Berechtigungsprofile mit frei geschalteten TRANSFER-ADMISSIONS einzurichten. Das Privileg wird im Member FTACADM der Parameterbibliothek definiert.

**Synchroner Auftrag**
synchronous request
: Die Benutzertask (Benutzerprozess, von der der *FT-Auftrag* abgegeben wurde, wartet auf das Ende der Übertragung. Der Benutzer kann nicht weiterarbeiten (vgl. *asynchroner Auftrag*).

**SYSFILE-Umgebung**
SYSFILE environment
: *Systemdateien* in BS2000-Systemen; als SYSFILE-Umgebung kann die Gesamtheit der einem Auftrag zugewiesenen Systemdateien bezeichnet werden.

**System**
system
: Siehe *FT-System*

**System, fernes**
system, remote
: Siehe *fernes System*

**System, lokales**
system, local
: Siehe *lokales System*

**Systemdateien**
system files
: Einem Auftrag in einem BS2000-System zugewiesene System-Ein-/Ausgabedateien. Auf Systemdateien kann der Benutzer nur indirekt zugreifen mit Hilfe des SYSFILE-Kommandos. Systemdateien stellen Daten und Hilfsmittel bereit, die die Funktionen des Organisationsprogramms benötigen.
  Systemdateien und ihre Primärzuweisungen:
  – SYSOUT: Ausgabe von Systemmeldungen an Datenstation
  – SYSLST: Ausgabe von Übersetzungsprotokollen etc. über Drucker (automatischer SPOOLOUT)
  – SYSLSTnn: wie SYSLST; $1 \leq nn \leq 99$; jede der bis zu 99 Systemdateien muss einer katalogisierten Datei zugeordnet sein
  – SYSOPT: Ausgabedatei wie SYSLST

– SYSCMD: dient der Kommandoangabe an das Organisationsprogramm
– SYSDTA: dient der Eingabe von Daten oder Anweisungen

**Systemglobale Benutzerverwaltung**
global user administration

Sie umfasst in BS2000-Systemen die Verwaltung von Benutzerkennungen und Benutzergruppen bezüglich Betriebsmitteln und Benutzerrechten, das Neueinrichten, Modifizieren und Löschen von Benutzerkennungen und Benutzergruppen.

**Systemglobale Privilegien**
global privileges

Alle mit dem BS2000-Kommando SET-PRIVILEGE vergebbaren Rechte sowie das Recht des Sicherheitsbeauftragten und das Recht der Kennung TSOS. *Systemglobale Privilegien* und *Systemverwalterrechte* sind identisch.

**Systemressourcen**
system resources

Ein Betriebsmittel eines Rechnersystems, das von einem *Job* oder einer *Task/Prozess* angefordert bzw. freigegeben werden kann.

**Systemverwalterkommando**
system administrator command

Kommando, das nicht unter einer beliebigen Benutzerkennung gegeben werden kann, sondern nur unter Benutzerkennungen, denen das entsprechende (systemglobale) Recht zugeordnet ist (in BS2000-Systemen z.B. unter der Benutzerkennung TSOS).

**Systemverwalterrechte**
system administrator privileges

Siehe *Systemglobale Privilegien*

**Systemverwaltung**
system administration

– Struktureinheit im Rechenzentrum
– Personenkreis, der Benutzerkennungen verwendet, an die systemglobale Rechte gebunden sind.

**Task**
task

Träger von Prozessen. Im BS2000 werden Tasks u.a. zum Abwickeln von Benutzerjobs (z.B. Batchjobs, Dialogjobs) eingesetzt, siehe *Job*.
In z/OS Instanz zur Ausführung eines oder mehrerer Programme innerhalb eines *Jobs*.

**TCP/IP (Transmission Control Protocol / Internet Protocol)**

Verbreitetes Protokoll zur Datenübertragung (entspricht etwa den Schichten 3 und 4 des *ISO/OSI-Referenzmodells*, d.h. Netzwerk- und Transportschicht). Wurde ursprünglich für das ARPANET (Rechnernetz des US-Verteidigungsministeriums) entwickelt, inzwischen de-facto-Standard.

**Top Secret**

Produkt der Firma Computer Associates für die Zugangs- und Zugriffssicherung.

**TranSON**

TranSON ist ein Software Produkt, das einen gesicherten Zugang zu einem Server ermöglicht. Der Einsatz von TranSON ist für die Anwendung transparent. Die Verbindung zum fernen Partner wird vom Arbeitsplatz über einen Client Proxy und Server Proxy zum fernen Partner geleitet. Der Client Proxy befindet sich auf dem Arbeitsplatz, der Server Proxy beim fernen Partner. Die Übertragung zwischen Client Proxy und Server Proxy ist verschlüsselt.

**Transparenter Modus**

transparent mode

Dateinamen, Vor-, Nach- und Folgeverbeitungen sowie ferne Kommandos, werden in einer festen binären Verschlüsselung unabhängig von lokalen Zeichencode-Einstellungen gesehen. Eine Codeumsetzung erfolgt lediglich zwischen EBCDIC DF.04-1 (BS2000), IBM1047 (z/OS) und ISO8859-1 (Unix, Windows).

**Transport Name Service (TNS)**

Dienst auf Unix- und Windows-Systemen zur Verwaltung transportsystemspezifischer Eigenschaften. Einträge für *Partnersysteme* enthalten die Informationen zum jeweils verwendeten *Transportsystem*.

**Transportprotokoll**

transport protocol

*Protokoll* der *Transportschicht*

**Transportschicht**

transport layer

Schicht 4 des *ISO/OSI-Referenzmodells*, wickelt die Protokolle für den Datentransport ab.

**Transport-Selektor (T-Selektor)**

transport selector

Subadresse, mit der eine ISO-8072-Anwendung in der *Transportschicht* angesprochen wird.

**Transportsystem**
transport system
- Teil eines Systems oder einer Architektur, der ungefähr die Funktionen der unteren vier OSI-Schichten erbringt, also den Transport der Nachrichten von einem Partner zum anderen Partner einer Kommunikationsbeziehung.
- Summe von Hardware- und Softwareeinrichtungen, die für den Datentransport in Rechnernetzen sorgt.

**TSN (Task Sequence Number)**
Identifikation eines BS2000-Prozesses (*Task*).

**Transportverbindung**
transport connection
Logische Verbindung zwischen zwei Benutzern des Transportsystems (Datenstationen oder Anwendungen).

**Übertragungseinheit**
transfer unit
In einer FTAM-Übertragung die kleinste Dateneinheit zum Transport von Dateiinhalten. Für *FTAM-1* und *FTAM-3* sind dies *Strings*. Eine Übertragungseinheit kann, muss aber nicht einem Satz der Datei entsprechen.

**Unicode**
Universelle Zeichencodierung, wird vom Unicode-Konsortium überwacht und gepflegt. Dieser Codierungs-Standard liefert die Grundlage, um Textdaten in beliebigen Sprachen mit moderner Software und IT-Protokollen zu verarbeiten, zu speichern und auszutauschen. Der Unicode-Standard definiert die drei Unicode Varianten UTF-8, UTF-16 und UTF-32.

**universal-class-number**
universal-class-number
Parameter des *document-type*, mit dem das *character-repertoire* einer zu übertragenden Datei festgelegt wird.

**UNIX®**
Eingetragenes Warenzeichen der Open Group für ein weit verbreitetes Mehrbenutzer-Betriebssystem. Ein System darf nur den Namen UNIX führen, wenn es von der Open Group zertifiziert ist.

**Unix-System**
Unix system
Allgemein übliche Bezeichnung für ein Betriebssystem, welches UNIX®-typische Funktionen implementiert und entsprechende Schnittstellen anbietet. Auch POSIX und Linux werden zu den Unix-Systemen gerechnet.

**Verarbeitungsrechner / Mainframe**
mainframe

Rechner (bestehend aus einem oder mehreren Prozessoren), der unter der Steuerung eines universellen Betriebssystems (z.B. BS2000 oder z/OS) arbeitet.
Synonym: BS2000-Rechner, Host-Rechner.

**Virtueller Dateispeicher**
virtual filestore

Im virtuellen FTAM-Dateispeicher stellt ein *FT-System* in der Rolle des *Responders* seine Dateien für *Partnersysteme* zur Verfügung. Die Darstellung einer Datei im virtuellen Dateispeicher ist durch die FTAM-Norm vorgegeben, siehe *Dateiattribute*.

**VisibleString**
visibleString

*character repertoire* für Dateien, die mit *FTAM-Partnern* übertragen werden.

**Volume-Set**
volume set

Bestandteil eines SM-Pubsets in einem BS2000-System. Ein Volume-Set ist eine Menge von Platten, die in den wesentlichen Eigenschaften (Plattenformat, Allokierungseinheit) übereinstimmen.
Der Name des Volume-Sets wird in einem Verzeichnis des SM-Pubsets verwaltet. Die Adressierung der Daten auf einem Volume des Volume-Sets wird jedoch über die SM-Pubset-Kennung realisiert.

**Vorrechner**
preprocessor / communication controller

Ein dem Verarbeitungsrechner (BS2000- oder z/OS-System) vorgelagertes Prozessorsystem, das spezielle Kommunikationsfunktionen im Netz wahrnimmt. Synonym: Kommunikationsrechner

**Vorverarbeitung**
preprocessing

Über die Vorverarbeitung erlaubt openFT das Abschicken eines Empfangsauftrags, bei dem nicht eine ferne Datei, sondern die Ausgaben eines fernen Kommandos bzw. Programms übertragen werden. Mit Hilfe der Vorverarbeitung sind z.B. Datenbankabfragen im fernen System möglich. Die Vorverarbeitung ist auch lokal möglich.

**VSAM**

Dateizugriffsmethode der IBM für sequentielle, direkte und indizierte Zugriffe.

**VTAM**

DFÜ-Zugriffsmethode der IBM.

**WAN (Wide Area Network)**

Öffentliches oder privates Netz, das große Entfernungen überbrücken kann und dabei - im Gegensatz zu *LAN*s - relativ langsam mit höherer Fehlerrate arbeitet. Heutzutage sind diese Definitionen nur noch eingeschränkt gültig, Beispiel: bei ATM-Netzen.

**Wiederanlauf**

restart

Automatische Fortsetzung eines *FT-Auftrags* nach einer Unterbrechung.

**Wiederanlaufpunkt**

restart point

Stelle, bis zu der die Daten der *Sendedatei* bei einer Unterbrechung der Dateiübertragung in der *Empfangsdatei* gesichert abgespeichert sind und ab der die Daten nach einem *Wiederanlauf* weiter übertragen werden.

**X.25**

Standardisierte Protokollfamilie für großräumige Computernetzwerke (WANs) über das Telefonnetz. Das Schichtmodell des Standards entspricht den Schichten 1-3 des OSI-Modells (Bitübertragung, Sicherungsschicht, Vermittlungsschicht).

**X-Terminal**

X terminal

In Unix-Systemen: Ein Bildschirm oder eine Softwarekomponente zur Darstellung der grafischen X Window-Oberfläche von Unix-Systemen. Ein X-Terminal oder eine entsprechende Software-Emulation ist Voraussetzung für den Einsatz der grafischen Oberfläche von openFT, dem openFT Explorer.

**Zeichenmodus**

character mode

Dateinamen, Vor-, Nach- und Folgeverbeitungen sowie ferne Kommandos werden in ihrer Zeichendarstellung gesehen. Ein Ä in einem fernen Dateinamen wird beispielsweise im Zeichenmodus auch im Partnersystem als Ä verstanden, auch wenn dort eine andere Systemcodierung eingerichtet bzw. eine andere lokale Codierung eingestellt ist.

**Zentrale Administration**
central administration

Die zentrale Administration von openFT umfasst die Funktionen *Fernadministration* und *ADM-Traps* und setzt den Einsatz eines *Fernadministrations-Servers* voraus.

**Zugangsberechtigung**
transfer admission

Kurzbezeichnung für *FTAC-Zugangsberechtigung*.

**Zugangsschutz**
access protection

Beinhaltet alle Methoden zum Schutz eines Datenverarbeitungssystems vor unberechtigtem Systemzugang.

**Zugriffsrecht / Zugriffsberechtigung**
access right / access admission

Leitet sich von der *Zugangsberechtigung* ab. Das Zugriffsrecht legt fest, worauf ein Benutzer, der die Zugangsberechtigung angegeben hat, Zugriff hat.

# Abkürzungen

| | |
|---|---|
| **ACSE** | Association Control Service Element |
| **AES** | Advanced Encryption Standard |
| **ANSI** | American National Standards Institute |
| **API** | Application Programming Interface |
| **API/CS** | Application Programming Interface/Communication System |
| **APPC** | Advanced Program-to-Program Communication |
| **APPN** | Advanced Peer-to-Peer Networking |
| **ARP** | Address Resolution Protocol |
| **ASCII** | American Standard Code for Information Interchange |
| **ASECO** | Advanced Security Control |
| **ASN** | Abstract Syntax Notation |
| **ATM** | Asynchronous Transfer Mode |
| **BCAM** | Basic Communication Access Method |
| **BSI** | Bundesamt für Sicherheit in der Informationstechnik |
| **CAE** | Common Application Environment |
| **CCP** | Communication Control Programm |
| **CCS** | Coded Character Set |
| **CCSN** | Coded Character Set Name |
| **CDDI** | Copper Distributed Data Interface |
| **CEN** | Comité Européen de Normalisation |
| **CENELEC** | Comité Européen de Normalisation Électrotechnique |
| **CICS** | Customer Information Control System (IBM) |
| **CMX** | Communication Manager Unix Systems |
| **COM** | Communication Port (asynchronous) |
| **CPX** | Compact Packet Exchange |
| **CSV** | Character Separated Values |
| **DAS** | Data Access Service |

| | |
|---|---|
| **DAP** | Directory Access Protocol |
| **DBA** | Data Base Access Service |
| **DCAM** | Data Communication Access Method |
| **DCE** | Data Communication Equipment |
| **DCE** | Distributed Computing Environment (OSF) |
| **DCM** | Data Communication Method |
| **DES** | Data Encryption Standard |
| **DFR** | Document File Retrieval |
| **DFS** | Distributed File System (DCE) |
| **DIN** | Deutsches Institut für Normung |
| **DME** | Distributed Management Environment |
| **DMS** | Data Management System (BS2000) |
| **DNS** | Domain Name Service |
| **DOS** | Disk Operating System |
| **DSA** | Directory System Agent |
| **DSC** | Data Stream Compatibility |
| **DSM** | Distributed Systems Management |
| **DSP** | Directory System Protocol |
| **DSSM** | Dynamic Subsystem Management |
| **DTE** | Data Termination Equipment |
| **DTS** | Distributed Time Service |
| **DVS** | Datenverwaltungssystem (BS2000) |
| **EBCDIC** | Extended Binary-Coded Decimal Interchange Code |
| **EN** | European Norm |
| **ENV** | Europäischer Normen-Vorschlag |
| **EPHOS** | European Procurement Handbook for Open Systems |
| **ERMS** | Entity Relationship Management System |
| **ES** | End System |
| **ETSI** | European Telecommunication Standards Institute |
| **EWOS** | European Workshop for Open Systems |
| **FADU** | File Access Data Unit |
| **FDDI** | Fiber Distributed Data Interface |
| **FEP** | Front End Processor |

| | |
|---|---|
| **FJAM** | File Job Access Method |
| **FMLI** | Form and Menu Language Interpreter |
| **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) |
| **FTP** | File Transfer Protocol |
| **FTPS** | FTP über SSL / TLS |
| **GOSIP** | Government OSI Profile |
| **GPL** | Gnu Public License |
| **GSM** | Global System for Mobile Communication |
| **HDLC** | High Level Data Link Control (ISO 7776) |
| **HNC** | Highspeed Net Connect |
| **HPFS** | High Performance File System |
| **HTTP** | Hypertext Transfer Protocol |
| **IBM** | International Business Machines Corporation |
| **ICC** | Intelligent Communication Controller |
| **ICMP** | Internet Control Message Protocol |
| **IEC** | International Electrotechnical Commission |
| **IEEE** | Institute of Electrical and Electronic Engineers |
| **IGMP** | Internet Group Management Protocol |
| **IMS** | Information Management System (IBM) |
| **IP** | Internet Protocol |
| **ISAM** | Index Sequential Access Method |
| **ISDN** | Integrated Services Digital Network |
| **ISO** | International Organization for Standardization |
| **IT** | Information Technology |
| **ITSEC** | Information Technology Security Evaluation Criteria (Europe, White Book) |
| **ITU** | International Telecommunication Union |
| **JCL** | Job Control Language |
| **LAN** | Local Area Network |
| **LMS** | Library Maintenance System |

| | |
|---|---|
| **LU** | Logical Unit |
| **MAC** | Medium Access Control |
| **MAN** | Metropolitan Area Network |
| **MCR** | Magnetic Card Reader |
| **MIB** | Management Information Base |
| **MLC** | Modular LAN Connect |
| **MSV** | Mittelschnelles Synchron Verfahren |
| **MVS** | Multiple Virtual System |
| **NCP** | Network Control Program (SNA) |
| **NCS** | Network Control System |
| **NDMS** | Network Data Management System |
| **NEA** | (Name der TRANSDATA-Architektur von Siemens) |
| **NFS** | Network File System |
| **NIS** | Network Information Service |
| **NTP** | Network Time Protocol |
| **ODI** | Open Data Link Interface |
| **ODI** | Open Device Interface |
| **ODL** | Object Description Language |
| **OSI** | Open Systems Interconnection |
| **OSS** | OSI Session Service |
| **PAM** | Pluggable Authentication Modules |
| **PC** | Personal Computer |
| **PCMX** | Portable Communication Manager Unix Systems |
| **PDU** | Protocol Data Unit |
| **PEM** | Privacy Enhanced Mail |
| **PICS** | Protocol Implementation Conformance Statement |
| **PIN** | Personal Identification Number |
| **PKCS** | Public Key Cryptography Standards |
| **PLAM** | Primary Library Access Method |
| **POP** | Post Office Protocol |
| **POSIX** | Portable Operating System Interface for Open Systems |
| **PSDN** | Packet Switched Data Network |
| **PU** | Physical Unit |

| | |
|---|---|
| **RFC** | Request for Comments |
| **RFC1006** | Request for Comments 1006 |
| **RJE** | Remote Job Entry |
| **RPC** | Remote Procedure Call |
| **RMS** | Reliant Monitor Services |
| **RTS** | Reliable Transfer Service |
| **SAM** | Sequential Access Method |
| **SAP** | Server Advertising Protocol (NetWare) |
| **SAP** | Service Access Point (OSI) |
| **SCM** | Software Configuration Management |
| **SDF** | System Dialog Facility |
| **SDLC** | Synchronous Data Link Control |
| **SESAM** | System zur Elektronischen Speicherung Alphanumerischer Merkmale |
| **SMF** | Service Management Facility (Solaris) |
| **SMF** | System Management Facility (IBM) |
| **SMTP** | Simple Mail Transfer Protocol |
| **SNA** | Systems Network Architecture |
| **SNMP** | Simple Network Management Protocol |
| **SNPA** | Subnetwork Point of Attachment |
| **SQL** | Structured Query Language |
| **SSL** | Secure Socket Layer |
| **TCP** | Transmission Control Protocol |
| **TCP/IP** | Transmission Control Protocol/Internet Protocol |
| **TELNET** | Telecommunications Network Protocol |
| **TFTP** | Trivial File Transfer Protocol |
| **TID** | Transport Identification |
| **TLS** | Transport Layer Security |
| **TNSX** | Transport Name Service in Unix systems |
| **TPI** | Transport Protocol Identifier |
| **TS** | Transport System |
| **UDP** | User Datagram Protocol |
| **UDS** | Universelles Datenbanksystem |
| **URL** | Uniform Resource Locator |

| | |
|---|---|
| **UTM** | Universal Transaction Monitor |
| **VDE** | Verband deutscher Elektrotechniker |
| **WAN** | Wide Area Network |
| **WS** | Workstation |
| **XDR** | External Data Representation |
| **XDS** | API to Directory Service |

# Literatur

Die Handbücher finden Sie im Internet unter *http://manuals.ts.fujitsu.com*. Handbücher, die mit einer Bestellnummer angezeigt werden, können Sie auch in gedruckter Form bestellen.

## Dokumentation zu openFT

**openFT**
**Konzepte und Funktionen**
Benutzerhandbuch

**openFT (BS2000)**
**Installation und Betrieb**
Systemverwalterhandbuch

**openFT (BS2000)**
**Kommandoschnittstelle**
Benutzerhandbuch

**openFT (BS2000)**
**Programmschnittstellen**
Programmierhandbuch

**openFT (Unix- und Windows-Systeme)**
**Installation und Betrieb**
Systemverwalterhandbuch

**openFT (Unix- und Windows-Systeme)**
**Kommandoschnittstelle**
Benutzerhandbuch

**openFT (Unix- und Windows-Systeme)**
**C- und Java-Programmschnittstelle**
Programmierhandbuch

**openFT (Unix- und Windows-Systeme)**
**openFT-Script-Schnittstelle**
Programmierhandbuch

**openFT (z/OS)**
**Installation und Betrieb**
Systemverwalterhandbuch

**openFT (z/OS)**
**Kommandoschnittstelle**
Benutzerhandbuch

## Dokumentation zum BS2000-Umfeld

**openNet Server** (BS2000)
**BCAM**
Benutzerhandbuch

**SNMP Management**
**SNMP Management für BS2000/OSD**
Benutzerhandbuch

**BS2000 OSD/BC**
**Kommandos** (mehrere Bände)
Benutzerhandbuch

**BS2000 OSD/BC**
**Makroaufrufe an den Ablaufteil**
Benutzerhandbuch

**IMON** (BS2000)
**Installationsmonitor**
Benutzerhandbuch

**BS2000 OSD/BC**
**Einführung in das DVS**
Benutzerhandbuch

**BS2000/OSD-BC**
**Verwaltung von Subsystemen (DSSM/SSCM)**
Benutzerhandbuch

**BS2000 OSD/BC**
**Systeminstallation**
Benutzerhandbuch

**BS2000 OSD/BC**
**Einführung in die Systembetreuung**
Benutzerhandbuch

**JV** (BS2000)
**Jobvariablen**
Benutzerhandbuch

**SECOS** (BS2000)
**Security Control System**
Benutzerhandbuch

**XHCS** (BS2000)
**8-bit-Code- und Unicode-Unterstützung im BS2000/OSD**
Benutzerhandbuch

**RAV** (BS2000)
**Rechenzentrums-Abrechnungs-Verfahren**
Benutzerhandbuch

**HIPLEX AF** (BS2000)
**Hochverfügbarkeit von Anwendungen in BS2000/OSD**
Produkthandbuch

## Dokumentation zum Umfeld von Unix-Systemen

**CMX**
**Betrieb und Administration**
Benutzerhandbuch

**CMX**
**Anwendungen programmieren**
Programmierhandbuch

**OSS(SINIX)**
**OSI Session Service**
User’s Guide

# Stichwörter

**T**

**X**



**Z**