

openFT V12.0 für Unix- und Windows-Systeme

openFT-Script-Schnittstelle

Benutzerhandbuch

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

Copyright © Fujitsu Technology Solutions GmbH 2012.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

Alle verwendeten Hard- und Softwarenamen sind Handelsnamen und/oder Warenzeichen der jeweiligen Hersteller.

Inhalt

1	Einleitung	7
1.1	Zielgruppe	7
1.2	Handbuchreihe von openFT für Unix- und Windows-Systeme	8
1.3	Änderungen gegenüber der Vorgängerversion	10
1.4	Darstellungsmittel	10
1.5	Readme-Datei	11
1.6	Voraussetzungen für openFT-Script	11
2	Aufbau eines Ftscripts	13
2.1	Bausteine eines Ftscripts	13
2.1.1	Aktivitäten	14
2.1.2	Kontext	15
2.1.3	Referenzierung	15
2.2	Datei- und Verzeichnisnamen angeben	19
2.2.1	Dateinamen-Attribute	20
2.2.2	Verzeichnisnamen-Attribute	21
2.3	Fehlerbehandlung	22
2.3.1	„Normale“ Ftscript-Fehlercodes	23
2.3.2	„Schwere“ Ftscript-Fehlercodes	24
2.3.3	Wiederanlauf	25
2.4	Ausführung eines Ftscripts	26

3	openFT-Script-Kommandos	29
3.1	Übersicht über die openFT-Script-Kommandos	29
3.2	ftcans - openFT-Script-Auftrag abbrechen	30
3.3	ftdels - openFT-Script-Auftrag löschen	32
3.4	ftmodsuo - openFT-Script-Benutzeroptionen ändern	34
3.5	ftshwsuo - openFT-Script-Benutzeroptionen anzeigen	36
3.6	ftscript - openFT-Script-Auftrag starten	38
3.7	ftshwact - Aktivitäten eines openFT-Script-Auftrags anzeigen	40
	Beschreibung der Ausgabe	42
3.8	ftshws - openFT-Script-Auftrag anzeigen	47
4	openFT-Script-Anweisungen	51
4.1	Syntax der openFT-Script-Anweisungen	51
4.2	baseDir	52
4.3	comment	53
4.4	context	54
4.5	createDirectory	55
4.6	deleteDirectory	57
4.7	deleteFile	59
4.8	directory	61
4.9	empty	62
4.10	executeScript	63
4.11	fault	66
4.12	faulthandler	69
	case	71
	default	71
4.13	file	73
4.14	foreach	74
4.15	ftscript	78

4.16	list	79
4.17	listDirectory	80
4.18	parallel	83
4.19	partner	85
4.19.1	processingAdmission	86
4.19.2	transferAdmission	87
	ftacAdmission	87
	userAdmission	87
4.20	script	89
4.21	sequence	91
4.22	transferFile	93
	autoDataSpec	99
	fromLocalFile	102
	fromLocalTmpFile	104
	fromRemoteFile	107
	remoteFailureScript	109
	remoteSuccessScript	111
	toLocalFile	112
	toLocalTmpFile	113
	toRemoteFile	115
5	Fehlermeldungen	117
	Fachwörter	141
	Stichwörter	145

1 Einleitung

Das Dokument beschreibt die Version 1 von openFT-Script.

openFT-Script ist eine Sprache zur Beschreibung von mehreren logisch voneinander abhängigen openFT-Aufträgen. Diese Aufträge können mit openFT-Script zu einem Auftrag (Ftscript) zusammengefasst werden.

openFT-Script entlastet den Kunden beim Überwachen zusammenhängender openFT-Aufträge und bietet einen Wiederanlauf im Fall einer Unterbrechung.

openFT-Script verwendet die XML-Notation.

Sie können openFT-Script unter Windows™ sowie unter Unix-Systemen verwenden.

1.1 Zielgruppe

Dieses Handbuch wendet sich an XML-Programmierer, die openFT-Script-Aufträge erstellen wollen. Durch openFT-Script-Aufträge werden openFT-Aufträge in Windows- oder Unix-Systemen gestartet und z.B. Dateien an oder von anderen Systemen übertragen.

Zum Verständnis des Handbuchs sind Kenntnisse von Windows und Unix-basierten Betriebssystemen sowie XML-Kenntnisse nützlich.

Das Handbuch ist gültig für Windows-Systeme, Solaris-Systeme sowie für die Portierungen auf andere Unix-Plattformen. Die betriebssystemabhängigen Unterschiede werden ausführlich in der Freigabemitteilung beschrieben, die auf der jeweiligen Produkt-CD mit ausgeliefert wird.

1.2 Handbuchreihe von openFT für Unix- und Windows-Systeme

Die vollständige Beschreibung von openFT und seinen Komponenten umfasst mehrere Handbücher. Neben dem vorliegenden Handbuch existieren weitere openFT-Handbücher für Unix- und Windows-Systeme.

Die Beschreibung ist folgendermaßen auf die Handbücher verteilt:

- openFT für Unix-Systeme - Installation und Administration
Das Systemverwalterhandbuch richtet sich an FT-, FTAC- und ADM-Verwalter. Es beschreibt:
 - die Installation von openFT und seinen optionalen Komponenten
 - Betrieb, Steuerung und Überwachung des FT-Systems und der FTAC-Umgebung
 - die Administrationskommandos für FT- und FTAC-Verwalter
 - die Konfiguration und den Betrieb eines Fernadministrations-Servers und eines ADM-Trap-Servers
 - wichtige CMX-Kommandos
- openFT für Windows-Systeme - Installation und Administration
Das Systemverwalterhandbuch richtet sich an FT-, FTAC- und ADM-Verwalter. Es beschreibt:
 - die Installation von openFT und seinen optionalen Komponenten
 - Betrieb, Steuerung und Überwachung des FT-Systems und der FTAC-Umgebung
 - die Administrationskommandos für FT- und FTAC-Verwalter
 - die Konfiguration und den Betrieb eines Fernadministrations-Servers und eines ADM-Trap-Servers
- openFT für Unix-Systeme - Managed File Transfer in der offenen Welt
Das Benutzerhandbuch richtet sich an den openFT-Benutzer und beschreibt:
 - die grundsätzlichen Funktionen der openFT-Produktfamilie
 - die Konventionen für den File Transfer zu Rechnern mit verschiedenen Betriebssystemen
 - Details zur Realisierung von FTAM
 - die openFT-Benutzerkommandos
 - die BSFT-Schnittstelle
 - die openFT-Script-Kommandos
 - Meldungen der verschiedenen Komponenten

- openFT für Windows-Systeme - Managed File Transfer in der offenen Welt
Das Benutzerhandbuch richtet sich an den openFT-Benutzer und beschreibt:
 - die grundsätzlichen Funktionen der openFT-Produktfamilie
 - die Konventionen für den File Transfer zu Rechnern mit verschiedenen Betriebssystemen
 - Details zur Realisierung von FTAM
 - die openFT-Benutzerkommandos
 - die openFT-Script-Kommandos
 - Meldungen der verschiedenen Komponenten
- openFT für Unix- und Windows-Systeme - Programmschnittstelle
Dieses Handbuch richtet sich an C-Programmierer und beschreibt die C-Programmschnittstelle auf Unix- und Windows-Systemen.
- openFT für Unix- und Windows-Systeme - openFT-Script-Schnittstelle
Dieses Handbuch richtet sich an XML-Programmierer und beschreibt:
 - die openFT-Script-Kommandos
 - Die XML-Anweisungen der openFT-Script-Schnittstelle



Viele der in den Handbüchern beschriebenen Funktionen sind auch über die grafische Oberfläche von openFT (openFT Explorer) verfügbar. Mit openFT Explorer wird eine ausführliche Online-Hilfe ausgeliefert, in der die Bedienung mit allen Dialogen beschrieben wird. Zusätzlich enthält diese Online-Hilfe auch die komplette Beschreibung der openFT-Kommandos.

1.3 Änderungen gegenüber der Vorgängerversion

Die openFT-Script-Schnittstelle zu openFT V12 bietet folgende neue Kommandos zum variablen Ablegen von openFT-Script-Aufträgen:

- *fmodsuo* zum Ändern von openFT-Script-Benutzeroptionen.
- *fishwsuo* zum Anzeigen von openFT-Script-Benutzeroptionen.

Die Kommandos *ficans*, *fidels* und *fishws* wurden geändert:

- *ftscriptid*: Zur Identifikation des openFT-Script-Auftrags können Sie in der *ftscriptid* die Wildcard-Symbole *?* und *** verwenden.

Der zugelassenen Speicherverbrauch zur Ausführung des Ftscripts in der Java Virtual Machine wurde mit openFT V10.0B20 erhöht, sodass auch umfangreiche openFT-Script-Aufträge mit hohen Speicheranforderungen ablaufen können. Gleichzeitig wurde auch der Ressourcenverbrauch durch parallele Threads reguliert (siehe [Abschnitt „parallel“ auf Seite 83](#)).

1.4 Darstellungsmittel

In diesem Handbuch werden folgende Darstellungsmittel verwendet:

`dicktengleiche Schrift`

Dicktengleiche Schrift wird für Eingaben und Beispiele verwendet.

kursive Schrift

Kursive Schrift wird im Fließtext verwendet, um Kommandos, Anweisungen, Namen, Variablen und Werte wie z.B. Dateinamen und Rechnernamen auszuzeichnen.



Das Symbol wird für Hinweistexte verwendet.

Für die Kommandobeschreibungen und die Programmschnittstelle werden zusätzliche Darstellungsmittel verwendet.

1.5 Readme-Datei

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

1.6 Voraussetzungen für openFT-Script

openFT-Script wird mit openFT ausgeliefert und setzt eine openFT-Version ab V10 auf dem ausführenden Rechner voraus. Alle adressierten Partner müssen ein FTAM-/openFT-kompatibles Produkt zur Dateiübertragung einsetzen.

Wird nicht openFT eingesetzt, so gelten die im openFT-Handbuch beschriebenen Einschränkungen.

Als Java-Laufzeitumgebung wird mindestens Java J2SE™ Runtime Environment 5.0 (JRE 5.0) benötigt. Auf Windows-Systemen ist außerdem die erweiterte Sprachversion (Unterstützung Nicht-Europäischer Sprachen, Extended Encoding Sets) des Java JRE notwendig. Diese muss nicht explizit installiert werden, wenn Java JDK oder Java ab 1.6 installiert ist.

2 Aufbau eines Ftscripts

2.1 Bausteine eines Ftscripts

Ein Ftscript besteht aus Aktivitäten. Jede Aktivität hat einen Kontext. In dem Kontext können auch Fehlerbehandlungen (*faulthandler*) beschrieben sein.

- **Aktivitäten** können Arbeitsanweisungen an openFT sein (z.B. *transferFile*, *deleteFile*) oder Anweisungen, die den Ablauf steuern (z.B. *parallel*, *foreach*). Die Anweisungen sind im [Kapitel „openFT-Script-Anweisungen“ auf Seite 51](#) beschrieben.
- Im **Kontext** können Dateien, Verzeichnisse, Scripts und Partner als Kontextobjekte hinterlegt werden. Durch **Referenzierung** können solche Kontextobjekte in der Aktivität und den darunterliegenden Aktivitäten, wenn diese kein Kontextobjekt mit dem gleichen Namen haben, mehrfach verwendet werden. Mehrfachverwendung ist außerhalb der Aktivität, in der das Kontextobjekt definiert ist, nicht möglich.
- Wenn innerhalb einer Aktivität ein Fehler (*fault*) auftritt, kann in der **Fehlerbehandlung** entsprechend reagiert werden. Wird die Fehlerbehandlung erfolgreich beendet (d.h. ohne Fehler), dann gilt auch die zugehörige Aktivität als erfolgreich beendet. Wenn die Fehlerbehandlung mit einem Fehler abbricht, dann gilt auch die Aktivität als fehlerhaft.

Der allgemeine Aufbau eines Ftscripts sieht wie folgt aus:

```
<ftscript version="1">
  <context>
    KontextObjekte*
    faulthandler?
  </context>
  Aktivität+
</ftscript>
```

Jedes Ftscript hat als Wurzel-Element die Anweisung `<ftscript version="1">`.

Das Wurzel-Element enthält folgende Unterelemente:

- einen (optionalen) Kontext mit Kontextobjekten und maximal einem *faulthandler*
- eine oder mehrere Aktivitäten, die in angegebener Reihenfolge ausgeführt werden.

Weitere Informationen zur Syntax siehe [Abschnitt „Syntax der openFT-Script-Anweisungen“ auf Seite 51](#).

2.1.1 Aktivitäten

Es gibt verschiedene Arten an Aktivitäten.

Interne Aktivitäten

Die internen Aktivitäten sind Anweisungen an den Ftscript-Interpreter, den Ablauf zu steuern.

Dazu gehören *ftscript*, *sequence*, *parallel*, *foreach*, *empty* und *fault* (Beschreibung siehe [Kapitel „openFT-Script-Anweisungen“](#) auf Seite 51).

Externe Aktivitäten

Die externen Aktivitäten sind Anweisungen an openFT, eine entsprechende Funktionalität auszuführen.

Dazu gehören *executeScript*, *transferFile*, *deleteFile*, *createDirectory*, *deleteDirectory* und *listDirectory* (Beschreibung siehe [Kapitel „openFT-Script-Anweisungen“](#) auf Seite 51).

Wenn Sie bei den externen Anweisungen *executeScript*, *deleteFile*, *createDirectory*, *deleteDirectory* oder *listDirectory* keinen Partner angeben, wird die Anweisung als lokales Kommando bzw. lokale Betriebssystemanweisung ausgeführt.

Eltern- und Kind-Aktivitäten

Sie können Aktivitäten schachteln (XML-Syntax). Deshalb werden Aktivitäten auch in Eltern- und Kind-Aktivitäten unterschieden.

```
<ftscript version="1">
  <parallel>
    <transferFile .../>
    <listDirectory .../>
    <foreach ...>
      <deleteFile .../>
    </foreach>
  </parallel>
</ftscript>
```

ftscript ist das Wurzel-Element.

Das Wurzel-Element hat ein Kind-Element (*parallel*).

parallel hat *ftscript* als Eltern-Element (auch „darüberliegendes Element“ genannt).

parallel hat außerdem drei Kind-Elemente (*transferFile*, *listDirectory* und *foreach*). Die *foreach* Aktivität hat wieder eine *deleteFile* Aktivität als Kind-Element.

2.1.2 Kontext

In dem Kontext einer Aktivität sind Kontextobjekte und Fehlerbehandlungen (*faulthandler*) beschrieben. Mit dem Sprachmittel Kontextobjekt können Sie im Ftscript ein Element, z.B. einen Partner, einmal spezifizieren und mehrfach verwenden. Hierzu referenzieren Sie das Kontextobjekt dort, wo es verwendet wird. Durch entsprechende Referenzierung können Sie auch die Eigenschaften mehrerer Kontextobjekte zusammensetzen.

Jedes Kontextobjekt hat eine Id, die innerhalb eines Kontextes eindeutig sein muss. Über diese Id wird das Kontextobjekt angesprochen (referenziert).

Wird ein referenziertes Kontextobjekt im aktuellen Kontext nicht gefunden, wird (rekursiv) in den darüberliegenden Kontexten gesucht. Kontextobjekte verdecken andere Kontextobjekte mit der gleichen Id in darüberliegenden Kontexten.

Ein Kontext ist immer vorhanden, auch wenn er nicht definiert wurde. Es gibt bestimmte Aktivitäten (*foreach, listDirectory*), die automatisch Daten in einen Kontext einblenden.

Die Kontextobjekte sind im [Abschnitt „context“ auf Seite 54](#) detaillierter beschrieben.

2.1.3 Referenzierung

Mit openFT-Script können Sie Kontextobjekte zusammensetzen.

Ein Partner (siehe [Abschnitt „partner“ auf Seite 85](#)) zum Beispiel muss nicht an jeder Stelle vollständig spezifiziert werden. Sie können einen Partner in ein Kontextobjekt auslagern und über Referenzierung mehrfach verwenden.

Für die Dereferenzierung gelten folgende Regeln:

1. Ein Attribut *ref* referenziert ein Kontextobjekt mit der spezifizierten Id und dem Objekttyp des Kontextobjekts (z.B. *file*, *partner*).
2. Die Suche wird im Eltern-Kontext weiter geführt, wenn im aktuellen Kontext kein passendes Objekt gefunden wird. Wird auch dort kein passendes Kontextobjekt gefunden, wird die Suche in dessen Eltern-Kontext fortgesetzt. Mehrstufige Referenzierungen sind erlaubt.
3. Wird kein passendes Kontextobjekt gefunden, wird das Script mit dem Fehler *ft_reference* abgebrochen (siehe [Abschnitt „Fehlerbehandlung“ auf Seite 22](#)).
4. *ref*="A" in einem Kontextobjekt mit der *ID*="A" ist erlaubt. *ref* bezieht sich in diesem Fall immer auf den Eltern-Kontext.
5. Zirkel sind nicht erlaubt:
obj1(*ID*="A", *ref*="B"); *obj2*(*ID*="B", *ref*="A") in demselben Kontext ist nicht möglich. Es wird der Fehler *ft_reference* ausgegeben.
6. Es werden alle Attribute und Elemente des referenzierten Elementes übernommen, die im referenzierenden Element nicht vorhanden sind. Vor dieser Übernahme wird das referenzierte Element seinerseits nach diesen Regeln dereferenziert.

Beispiele

1. Gültige Referenzierung

```
<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <context>
    <partner id="remote" name="WindowsP_1">
      <transferAdmission>
        <ftacAdmission ftacAdmission="FTACADM" />
      </transferAdmission>
    </partner>
    <file id="pack" name="pack1.bin">
      <partner ref="remote" />
      <directory name="frg_eis_01" />
    </file>
  </context>
  <transferFile>
    <fromRemoteFile ref="pack" />
    <toLocalFile name="pack1.bin">
      <directory name="frg_eis_01" />
    </toLocalFile>
  </transferFile>
  <transferFile>
    <context>
      <partner id="remote" name="UnixP_1">
        <transferAdmission>
          <ftacAdmission ftacAdmission="FTACADM" />
        </transferAdmission>
      </partner>
    </context>
    <fromRemoteFile ref="pack">
      <partner ref="remote" />
    </fromRemoteFile>
    <toLocalFile name="pack2.bin">
      <directory name="frg_eis_01" />
    </toLocalFile>
  </transferFile>
</ftscript>
```

Das Dateiojekt mit der *Id="pack"* referenziert ein Partnerobjekt mit der *Id="remote"*.

In der ersten *transferFile* Aktivität wird der Partner *WindowsP_1* verwendet.

In der zweiten *transferFile* Aktivität wird dasselbe Dateiojekt *pack* referenziert. Der Partner wurde jedoch überschrieben. Dadurch wird der Partner *UnixP_1* angesprochen, der im Kontext des zweiten *transferFile* definiert ist. Diese Definition überdeckt den Windows-Partner mit der *Id="remote"*.

Wenn der Partner nicht überschrieben wäre, würde der Windows-Partner verwendet werden, da dieser von der Objektdefinition *pack* im Kontext von *ftscript* gefunden wird. Der Partner (auf dem Unix-System) aus dem Kontext des zweiten *transferFile* würde ignoriert werden, da er von der Definition *Id="pack"* aus nicht mehr zugänglich ist.

2. Ungültige Referenzierung

```
<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <transferFile>
    <fromRemoteFile ref="pack"/>
    <toLocalFile name="pack1.bin">
      <directory name="frg_eis_02"/>
    </toLocalFile>
  </transferFile>
</ftscript>
```

Die Referenz *pack* ist nicht definiert. Das Script gibt einen Fehler *ft_noRef* aus. Dieser wird nicht behandelt (kein *faulthandler* definiert). Das Script wird vor Ausführung der *transferFile* Aktivität beendet. Ein Wiederanlauf ist nicht mehr möglich. Mit *ftshws* können Sie den Grund für den Abbruch anzeigen.

2.2 Datei- und Verzeichnisnamen angeben

Viele openFT-Script-Anweisungen verwenden Datei- oder Verzeichnisnamen. Diese werden in den Attributen *name*, *bs2000Name*, *unixName*, *windowsName* und *zosName* angegeben. Sofern keine Besonderheiten angegeben sind, gelten die folgenden Definitionen.

Wird ein Partner angegeben, so muss dessen Betriebssystem explizit angegeben werden. Das Betriebssystem des lokalen Rechners ermittelt openFT-Script selbst.

Regeln für Dateinamen- und Verzeichnisnamen-Attribute

- Ein betriebssystemspezifisches Dateinamen- oder Verzeichnisnamen-Attribut (*bs2000Name*, *unixName*, *windowsName* oder *zosName*) wird ausgewertet, wenn es keinen Leer-String enthält und wenn das angegebene beziehungsweise ermittelte Betriebssystem diesen Typ hat.
- Wird kein passendes betriebssystemspezifisches Namens-Attribut gefunden oder ist es mit Leerstring belegt, dann gilt stattdessen das nicht-betriebssystemspezifische Attribut *name*. Bitte beachten Sie, dass *name* als Standardwert den Leer-String besitzt.
- Wenn Sie als Zugangsberechtigung ein Profil verwenden, in dem ein Datei- oder Verzeichnisname definiert ist (siehe openFT-Benutzerhandbuch), dann ist es am einfachsten, wenn Sie alle Dateinamen- oder Verzeichnisnamen-Attribute weglassen. Damit wird der Leer-String verwendet (= Standardwert des betriebssystem-unspezifischen Attributs, siehe oben).

Alternative: Versorgen Sie alle relevanten betriebssystemspezifischen Dateinamen- oder Verzeichnisnamen-Attribute mit Leer-String. Das nicht-betriebssystemspezifische Attribut können Sie dann entweder weglassen oder ebenfalls mit Leer-String versorgen.

2.2.1 Dateinamen-Attribute

Wenn ein Dateiname mit dem Pipe-Zeichen („|“) beginnt, hat dies die Sonderbedeutung eines erzeugenden (*fromLocalFile*, *fromRemoteFile*) oder empfangenden (*toLocalFile*, *toRemoteFile*) Programmaufrufs (siehe openFT-Benutzerhandbuch).

Einschränkung: Dateinamen dürfen maximal 512 Zeichen lang sein. Diese Grenze wird bereits beim Stellen des openFT-Script-Auftrags geprüft (statische Prüfung). Je nach Betriebssystem können dies auch weniger Zeichen sein (siehe openFT-Benutzerhandbuch). Diese betriebssystemspezifische Länge wird erst zum Ablauf des Ftscripts geprüft (dynamische Prüfung).

Bitte beachten Sie den Abschnitt [„Regeln für Dateinamen- und Verzeichnisnamen-Attribute“ auf Seite 19](#).

Name	Wert	Bedeutung
name?	string " "	Dateiname mit Angabe des Teilpfades (siehe openFT-Benutzerhandbuch). <i>name</i> wird verwendet, wenn kein Betriebssystem bekannt ist oder kein betriebssystemspezifischer Name angegeben wurde. Der Standardwert ist ein Leer-String. Ist <i>name</i> nicht angegeben und das Profil <i>partner-ftac</i> mit einem festen Dateinamen verknüpft, so gilt dieser. Als Trennsymbol dürfen Sie nur das Zeichen „/“ verwenden (z.B. <i>/C:/x/y</i> in Unix-Systemen oder <i>C:/x/y</i> in Windows-Systemen).
bs2000Name?	string " "	Dateiname für BS2000. Der Name wird verwendet, wenn er angegeben wurde und der zugehörige Partner ein BS2000-System ist.
windowsName?	string " "	Dateiname für Windows. Der Name wird verwendet, wenn er angegeben wurde und der zugehörige Partner oder das zugehörige lokale System ein Windows-System ist. Sie können die „/“-Schreibweise oder die Windows-spezifische Schreibweise verwenden.
unixName?	string " "	Dateiname für Unix-System. Der Name wird verwendet, wenn er angegeben wurde und der zugehörige Partner oder das zugehörige lokale System ein Unix-System ist. Sie dürfen nur die „/“-Schreibweise verwenden.
zosName?	string " "	Dateiname für z/OS. Der Name wird verwendet, wenn er angegeben wurde und der zugehörige Partner ein z/OS-System ist.

2.2.2 Verzeichnisnamen-Attribute

Einschränkung: Verzeichnisnamen dürfen maximal 512 Zeichen lang sein. Diese Grenze wird bereits beim Stellen des openFT-Script-Auftrags geprüft (statische Prüfung). Je nach Betriebssystem können dies auch weniger Zeichen sein (siehe openFT-Benutzerhandbuch). Diese betriebssystemspezifische Länge wird erst zum Ablauf des Ftscripts geprüft (dynamische Prüfung).

Bitte beachten Sie den Abschnitt „[Regeln für Dateinamen- und Verzeichnisnamen-Attribute](#)“ auf Seite 19

Name	Wert	Bedeutung
name?	string " "	Verzeichnisname. <i>name</i> wird verwendet, wenn kein Betriebssystem bekannt ist oder kein betriebssystemspezifischer Name angegeben wurde. Der Standardwert ist ein Leer-String. Ist <i>name</i> nicht angegeben und das Profil <i>partner-ftac</i> mit einem festen Verzeichnisnamen verknüpft, so gilt dieser. Als Trennsymbol dürfen Sie nur das Zeichen „/“ verwenden.
bs2000Name?	string " "	BS2000-spezifische Adressierung (siehe openFT-Benutzerhandbuch).
windowsName?	string " "	Windows-spezifischer Pfad. Der Name wird verwendet, wenn er angegeben wurde und der zugehörige Partner oder das zugehörige lokale System ein Windows-System ist. Sie können die „/“-Schreibweise oder die Windows-spezifische Schreibweise verwenden.
unixName?	string " "	Unix-System spezifischer Pfad. Der Name wird verwendet, wenn er angegeben wurde und der zugehörige Partner oder das zugehörige lokale System ein Unix-System ist. Sie dürfen nur die „/“-Schreibweise verwenden.
zosName?	string " "	z/OS-spezifische Adressierung (siehe openFT-Benutzerhandbuch).

2.3 Fehlerbehandlung

Beim Einlesen wird das Ftscript einer Prüfung unterzogen.

Treten hierbei Fehler auf, wird das Ftscript nicht ausgeführt. Beim Starten des Ftscripts erhalten Sie eine entsprechende Fehlermeldung und einen Returncode (siehe [Abschnitt „ftscript - openFT-Script-Auftrag starten“ auf Seite 38](#)).

Bei erfolgreicher Prüfung wird das Ftscript asynchron ausgeführt. Treten hierbei Fehler auf, wird die Fehlerursache protokolliert.

Jede Fehlermeldung hat einen eindeutigen internen Code und ist einem bestimmten Fehlercode zugeordnet. Anhand des Fehlercodes werden die Fehler einer der beiden folgenden Kategorien zugeteilt:

- „normale“ Ftscript-Fehlercodes
- „schwere“ Ftscript-Fehlercodes

Die Tabelle aller Fehlermeldungen finden Sie im [Kapitel „Fehlermeldungen“ auf Seite 117](#).

Die von openFT-Script vergebenen Fehlercodes beginnen immer mit „ft_“.

Sie können zusätzlich über die *fault* Aktivität (siehe [Abschnitt „fault“ auf Seite 66](#)) beliebige andere Fehlercodes vergeben, die jedoch nicht mit „ft_“ beginnen dürfen. Diese gelten immer als „normale“ Fehlercodes.

Die Fehlerursache kann mit *ftshwact* angezeigt werden.

Wenn für einen Fehler kein *faulthandler* existiert, wird das *ftscript* mit einem Fehler (Status F) beendet. Der Status und die Ursache des Abbruchs kann mit *ftshws* angezeigt werden.

2.3.1 „Normale“ Ftscript-Fehlercodes

Die „normalen“ Fehler beziehen sich auf die zu übertragenden Objekte oder beteiligten Rechner.

Ein "normaler" Fehler kann mit dem *faulthandler (default)* abgefangen werden (siehe [Abschnitt „default“ auf Seite 71](#)).

Fehlercodes	Beschreibung
ft_access	Auf eine Datei/Verzeichnis/Rechner kann nicht zugegriffen werden.
ft_admin	Administrationsfehler.
ft_auth	Authentifizierungsfehler (falsche Kennung/Passwort/Rechte).
ft_cantCreate	Eine Datei/Verzeichnis kann nicht angelegt werden.
ft_cantDelete	Eine Datei/Verzeichnis kann nicht gelöscht werden.
ft_configuration	Konfigurationsfehler.
ft_connection	Ein Verbindungsfehler ist aufgetreten.
ft_corrupt	Eine Datei entspricht nicht dem erwarteten Format.
ft_exist	Eine Datei/Verzeichnis existiert schon.
ft_localFileStructure	Fehler in der lokalen Datei.
ft_notEmpty	Ein zu löschendes Verzeichnis ist nicht leer.
ft_notExist	Eine Datei/Verzeichnis/CCS/Dateieigentümer existiert nicht.
ft_remoteFileStructure	Fehler in der fernen Datei.

2.3.2 „Schwere“ Ftscript-Fehlercodes

Die „schweren“ Fehler werden vor allem durch interne Fehler (z.B. mangelnder Speicherplatz für Verwaltungsinformationen auf der Festplatte) oder Script-Fehler (z.B. nicht aufgelöste Referenzen) verursacht.

Tritt ein „schwerer“ Fehler auf, ist ein Wiederanlauf nur bedingt möglich (siehe [Abschnitt „Wiederanlauf“ auf Seite 25](#)).

Die „schweren“ Fehler können nicht durch den *faulthandler (default)* abgefangen werden. In diesem Fall muss ein expliziter *faulthandler (case)* geschrieben werden (siehe [Abschnitt „case“ auf Seite 71](#)).

Fehlercodes	Beschreibung
ft_abort	Abbruch durch den Anwender.
ft_error	Ein allgemeiner Fehler ist aufgetreten.
ft_notSupported	Die Funktionalität wird nicht unterstützt.
ft_panic	Ein schwerwiegender Fehler ist aufgetreten.
ft_paramError	Ein Parameter-Fehler ist aufgetreten.
ft_paramTooLong	Ein Parameter liegt außerhalb des Wertebereichs.
ft_recoveryFailed	Fehler beim Wiederanlauf (siehe Abschnitt „Wiederanlauf“ auf Seite 25). Der Fehler kann nicht abgefangen werden.
ft_reference	Eine Referenz ist ungültig (nicht vorhanden oder zirkulär).
ft_resource	Ein Ressourcen-Fehler ist aufgetreten (z.B. zu wenig Speicherplatz)
ft_syntax	Ein Syntaxfehler ist aufgetreten.

2.3.3 Wiederanlauf

Ftscripts sind im allgemeinen wiederanlauffähig, wenn sie z.B. aufgrund eines Systemabsturzes abgebrochen sind. Einschränkungen existieren nur bei folgenden Aktivitäten:

- *executeScript*, wenn *repeatable=no* angegeben wurde
- *createDirectory*, wenn *faultIfExists* angegeben wurde
- *deleteFile* oder *deleteDirectory*, wenn *faultIfNotExists* angegeben wurde

Wenn der openFT-Script-Auftrag genau während der Bearbeitung der Anweisung abgebrochen ist, kann nicht mehr entschieden werden, ob die Aktivität bereits ausgeführt wurde. In den oben genannten Fällen ist beim Wiederanlauf nicht eindeutig, wie das Ftscript weiter ausgeführt werden soll.

Existiert z.B. ein Verzeichnis schon, das angelegt werden sollte, lässt sich nicht mehr feststellen, ob es durch die abgebrochene *createDirectory* Aktivität angelegt wurde oder bereits vor dem Ablauf des openFT-Script-Auftrags existierte.

Trifft der Wiederanlauf eines Ftscripts auf eine solche „unklare“ Situation, so reagiert er folgendermaßen:

Aktivität	Verhalten beim Wiederanlauf
<i>executeScript</i> mit Attribut <i>repeatable=no</i>	Abbruch der Aktivität mit Fehler <i>ft_resumeForbidden</i>
<i>createDirectory</i> mit Attribut <i>faultIfExists</i>	Wenn das Verzeichnis schon existiert, Abbruch der Aktivität mit Fehler <i>ft_recoveryCreateDirectory</i>
<i>deleteFile</i> oder <i>deleteDirectory</i> mit Attribut <i>faultIfNotExists</i>	Wenn Datei/Verzeichnis nicht (mehr?) existiert, Abbruch der Aktivität mit Fehler <i>ft_recoveryFailed</i>

Dieses Verhalten kann auch auftreten, wenn die openFT-Instanz umgeschaltet wird. Beim Löschen der openFT-Instanz werden alle noch laufenden openFT-Script-Aufträge unterbrochen. Nach dem Umschalten der Instanz laufen sie wieder an. In den oben genannten Fällen wird mit dem Unterbrechen des Auftrags ca. 2 Sekunden auf das Ende der Aktivität gewartet. Bei einer langlaufenden *executeScript* Aktivität ist dies u.U. nicht ausreichend, so dass dieser openFT-Script-Auftrag beim Versuch des Wiederanlaufs mit Fehler abgebrochen wird.

2.4 Ausführung eines Ftscripts

Wenn ein Ftscript ausgeführt wird, durchläuft jede Aktivität folgende Zustände:

- Initialisierung
- Ausführung
- Ende
- (Fehler)

Initialisierung

Der Kontext wird bereitgestellt.

Ausführung

Bei externen Aktivitäten wird die openFT-Funktionalität ausgeführt.
Bei internen Aktivitäten wird die entsprechende Anweisung ausgeführt.

Tritt bei der Ausführung einer Aktivität ein Fehler auf, wird ein Fehler (*fault*) mit einem Fehlercode ausgegeben. Die Aktivität geht in den Zustand „Fehler“.

Ende

Das Ende der Aktivität wird erreicht, wenn die Ausführung ohne Fehler abgeschlossen werden konnte. Eventuell werden Daten in den darüberliegenden Kontext eingeblendet.

Die Aktivität wird terminiert. Es wird mit der nächsten Aktivität fortgefahren.
Gibt es keine weitere Aktivität, ist das Ftscript beendet.

Fehler

Der Zustand „Fehler“ kann hervorgerufen werden:

- durch einen Fehler bei der Ausführung der Aktivität selbst
- durch einen Fehler in einer Kind-Aktivität, die nicht durch einen *faulthandler* abgefangen wird

Im aktuellen Kontext wird ein passender *faulthandler* gesucht (siehe [Abschnitt „faulthandler“ auf Seite 69](#)). Die Aktivität wird durch den Inhalt des *faulthandler* ersetzt. Die Kontextobjekte der Aktivität werden dabei in den Kontext des *faulthandler* eingeblendet. Alle Kind-Aktivitäten der Aktivität werden abgebrochen und ihre Kontexte gehen verloren. Sind von diesen Kind-Aktivitäten laufende Aufträge zur Dateiübertragung gestartet worden, werden diese ebenfalls abgebrochen.

Wird kein passender *faulthandler* gefunden, wird der Fehler an die Eltern-Aktivität übergeben. Die Eltern-Aktivität geht in den Zustand „Fehler“.

Wird auch in keiner übergeordneten Aktivität ein passender *faulthandler* gefunden, wird das gesamte Ftscript terminiert.

Diagnose-Informationen

Zur genauen Analyse des Ftscript-Ablaufs einschließlich Wiederanlauf (siehe [Seite 25](#)) können Sie einen Trace aktivieren:

```
ftscript -t <Ftscriptdateiname>
```

Der Trace protokolliert jede Aktion des Auftrags.

3 openFT-Script-Kommandos

Die openFT-Script-Kommandos dienen dazu, openFT-Script-Aufträge zu starten und zu verwalten. Die Aufträge selbst werden in Form von XML-Anweisungen in einer Textdatei hinterlegt. Diese Anweisungen sind ab [Seite 51](#) beschrieben.

3.1 Übersicht über die openFT-Script-Kommandos

Starten und Beenden von openFT-Script-Aufträgen

<code>ftscript</code>	openFT-Script-Auftrag starten
<code>ftcans</code>	openFT-Script-Auftrag abrechnen
<code>ftdels</code>	openFT-Script-Auftrag löschen

Anzeigen von openFT-Script-Aufträgen und openFT-Script-Aktivitäten

<code>ftshws</code>	openFT-Script-Aufträge anzeigen
<code>ftshwact</code>	Aktivitäten eines openFT-Script-Auftrags anzeigen

Unter Unix-Systemen steht dem FT-Verwalter zusätzlich das Kommando *ftsetjava* zur Verfügung, um den Link auf das Java-Executable zu verwalten (nähere Informationen siehe openFT-Handbuch „Installation und Administration“).

Als FT-Verwalter können Sie alle openFT-Script-Aufträge im System ansehen, abrechnen und löschen sowie die Aktivitäten aller openFT-Script-Aufträge ansehen. Als Benutzer ohne Verwalterrechte können Sie nur eigene openFT-Script-Aufträge verwalten.

Variables Ablegen von openFT-Script-Aufträgen

<code>ftmodsuo</code>	openFT-Script-Benutzeroptionen ändern
<code>ftshwsuo</code>	openFT-Script-Benutzeroptionen anzeigen

3.2 ftcans - openFT-Script-Auftrag abbrechen

Mit *ftcans* können Sie noch nicht abgeschlossene openFT-Script-Aufträge eines Benutzers abbrechen. Sie können entweder einen bestimmten openFT-Script-Auftrag oder alle openFT-Script-Aufträge eines Benutzers abbrechen. Ggf. werden damit auch die vom angegebenen openFT-Script-Auftrag gestarteten und aktuell laufenden Dateiübertragungsaufträge abgebrochen. Dies kann eine gewisse Zeit in Anspruch nehmen. Anschließend wird der Status des openFT-Script-Auftrags auf "cancelled" gesetzt, so dass kein Wiederanlauf erfolgt.

Wird der abzubrechende openFT-Script-Auftrag gerade bearbeitet, wird im Anschluss die folgende Meldung auf *stderr* ausgegeben:

```
ftcans: Abbruchanforderung für ftscrip id ftsript id gestartet
```

Wenn der Auftrag gestartet ist, aber noch nicht bearbeitet wird, wird im Anschluss die folgende Meldung auf *stderr* ausgegeben:

```
ftcans: ftscrip id ftsript id abgebrochen
```

Format für Unix-Systeme

```
ftcans -h |
        [ -u=<Benutzerkennung 1..32> ]
        <ftsriptid> | @a
```

Format für Windows-Systeme

```
ftcans -h |
        [ -u=<Benutzerkennung 1..36> ]
        <ftsriptid> | @a
```

Beschreibung

-h gibt die Kommandosyntax am Bildschirm aus. Weitere Angaben nach *-h* werden ignoriert.

-u=Benutzerkennung

Benutzerkennung, unter der der abzubrechende openFT-Script-Auftrag gesucht wird.

Die Angabe einer Benutzerkennung ist nur für den FT-Verwalter erlaubt.

Standard ist die Benutzerkennung des Aufrufers.

ftscriptid

Identifikation des openFT-Script-Auftrags. Diese wird ausgegeben, wenn der openFT-Script-Auftrag per *ftscript*-Kommando gestartet wird.

In der *ftscriptid* können Sie die Wildcard-Symbole *?* und *** verwenden. Dann werden alle openFT-Script-Aufträge abgebrochen, die dem Wildcard-Muster entsprechen.

? wird als genau ein beliebiges Zeichen interpretiert.

*** wird als beliebig viele, beliebige Zeichen interpretiert.

Geben Sie bei Wildcard-Verwendung die *ftscriptid* in Hochkommas eingeschlossen an, damit die Wildcard-Symbole nicht durch die Shell interpretiert werden.

@*a* bedeutet, dass alle openFT-Script-Aufträge des Benutzers abgebrochen werden sollen.

Returncode

0	OK
4	Syntaxfehler
51	Fehler beim Ausgeben eines Ftscript-Benutzers
54	Ftscript Id nicht gefunden
250	Interner Fehler

3.3 ftdels - openFT-Script-Auftrag löschen

Der angegebene, beendete openFT-Script-Auftrag wird aus dem Verzeichnis des Benutzers gelöscht bzw. es werden alle beendeten openFT-Script-Aufträge aus dem Verzeichnis des Benutzers gelöscht.

Für gelöschte Aufträge sind danach keine Informationen mehr verfügbar. Ein *ftshws* oder *ftshwact*-Kommando mit der *ftscriptid* eines gelöschten Auftrags wird abgewiesen, da sie nicht mehr existiert.

Damit ein openFT-Script-Auftrag gelöscht werden kann, muss er beendet sein, d.h. *ftshws* muss den Status T, F oder C anzeigen.



Da das *ftcans*-Kommando nicht synchron ist, muss vor einem nachfolgenden *ftdels* ggf. auf das Eintreten des Zustandes C (cancelled) gewartet werden.

Wird kein *ftdels* für einen openFT-Script-Auftrag gegeben, so wird dieser bei Ablauf der Löschfrist von 3 Tagen automatisch gelöscht.

Format für Unix-Systeme

```
ftdels -h |
        [-u=<Benutzerkennung 1..32> ]
        <ftscriptid> | @a
```

Format für Windows-Systeme

```
ftdels -h |
        [-u=<Benutzerkennung 1..36> ]
        <ftscriptid> | @a
```

Beschreibung

-h gibt die Kommandosyntax am Bildschirm aus. Weitere Angaben nach *-h* werden ignoriert.

-u=Benutzerkennung

Benutzerkennung, unter der der zu löschende openFT-Script-Auftrag gesucht wird.

Die Angabe einer Benutzerkennung ist nur für den FT-Verwalter erlaubt.

Standard ist die Benutzerkennung des Aufrufers.

ftscriptid

Identifikation des openFT-Script-Auftrags. Diese wird ausgegeben, wenn der openFT-Script-Auftrag per *ftscript*-Kommando gestartet wird.

In der *ftscriptid* können Sie die Wildcard-Symbole ? und * verwenden. Dann werden alle openFT-Script-Aufträge gelöscht, die dem Wildcard-Muster entsprechen.

? wird als genau ein beliebiges Zeichen interpretiert.

* wird als beliebig viele, beliebige Zeichen interpretiert.

Geben Sie bei Wildcard-Verwendung die *ftscriptid* in Hochkommas eingeschlossen an, damit die Wildcard-Symbole nicht durch die Shell interpretiert werden.

@a bedeutet, dass alle beendeten openFT-Script-Aufträge des Benutzers gelöscht werden sollen.

Returncode

0	OK
4	Syntaxfehler
51	Fehler beim Ausgeben eines Ftscript-Benutzers
54	Ftscript Id nicht gefunden
56	openFT-Script-Auftrag ist nicht beendet
250	Interner Fehler

3.4 ftmodsuo - openFT-Script-Benutzeroptionen ändern

Ab openFT V12 können Sie als Benutzer spezifizieren, in welchem Verzeichnis Ihre openFT-Script-Aufträge abgelegt werden sollen. Im genannten Arbeitsverzeichnis legt openFT-Script das Unterverzeichnis *.openFT/<instanz>/script* bzw. *.openFT\<instanz>\script* an und die openFT-Script-Aufträge darin ab. Das Unterverzeichnis wird mit Schreibrecht für Sie versehen und ist für andere Benutzer nicht zugreifbar.

Mit dem Kommando *ftmodsuo* geben Sie das Verzeichnis an, in dem die openFT-Script-Aufträge abgelegt werden sollen. Dies ist jedoch nur zulässig, wenn kein openFT-Script läuft und keine aktuellen openFT-Script-Aufträge des Benutzers existieren. Gegebenenfalls müssen Sie Ihre laufenden openFT-Script-Aufträge mit *ftcans* abrechnen und beendete openFT-Script-Aufträge mit *ftdels* löschen. Das Kommando wird auch abgewiesen, wenn unter derselben Benutzerkennung gerade ein anderes *ftmodsuo*-Kommando zur Spezifizierung eines openFT-Script-Arbeitsverzeichnisses läuft.

Format

```
ftmodsuo -h |  
          [ -wd=[ <Verzeichnisname 1..128> ] ]
```

Beschreibung

- h** gibt die Kommandosyntax am Bildschirm aus. Weitere Angaben nach *-h* werden ignoriert.
- wd** absoluter oder relativer Pfadname des Arbeitsverzeichnisses, in dem das Unterverzeichnis für die openFT-Script-Aufträge des Benutzers angelegt werden soll.

Mit *-wd=* wird das Arbeitsverzeichnis zurückgesetzt auf den Standardwert, d.h. das Home-Verzeichnis des Benutzers.

ftmodsuo ohne Parameter ist zulässig, bewirkt jedoch nichts.

Returncode

- 0 OK
- 4 Syntaxfehler (z.B. der Name des Arbeitsverzeichnisses ist zu lang)
- 15 openFT hat keine Berechtigung für diesen Benutzer Aufträge zu bearbeiten (z.B. Kennwort nicht gesetzt bei Home-Verzeichnis-Zugriff)
- 69 Fehler beim Zugriff auf Datei (*Prelock.lck/UserLock.lck* in *FtscriptWorkdir*)
- 79 openFT-Script Interpreter oder anderes *ftmodsuo* läuft. Kommando abgebrochen
- 80 Aktuelle openFT-Script-Aufträge sind vorhanden. Kommando abgebrochen
- 81 Altes openFT-Script-Arbeitsverzeichnis nicht zugreifbar
- 88 Unterverzeichnisse im openFT-Script-Arbeitsverzeichnis können nicht angelegt werden.

Bedeutung: Das Verzeichnis *<wd>/openFT/<Instanzname>/script* bzw. *<wd>\openFT\<Instanzname>\script* konnte nicht angelegt werden, z.B. weil die Schreibzugriffsberechtigung fehlt oder ein physikalischer Fehler auftrat.
- 90 Arbeitsverzeichnis existiert nicht. Kommando abgebrochen
- 91 Warnung: Das bisherige Arbeitsverzeichnis konnte nicht überprüft werden

3.5 ftshwsuo - openFT-Script-Benutzeroptionen anzeigen

Mit dem Kommando *ftshwsuo* zeigen Sie das Verzeichnis an, in dem die openFT-Script-Aufträge abgelegt werden sollen.

Format für Unix-Systeme

```
ftshwsuo -h |
           [ -csv ]
           [ -u=<Benutzerkennung 1..32 | @a ]
```

Format für Windows-Systeme

```
ftshwsuo -h |
           [ -csv ]
           [ -u=<Benutzerkennung 1..36 | @a ]
```

Beschreibung

- h** gibt die Kommandosyntax am Bildschirm aus. Weitere Angaben nach *-h* werden ignoriert.
- csv** Die Informationen werden im CSV-Format ausgegeben. Wenn Sie *-csv* nicht angeben, werden die Informationen im Tabellenformat ausgegeben.
- u=Benutzerkennung | @a**
nur für den FT-Verwalter

Benutzerkennung, deren openFT-Script-Optionen angezeigt werden sollen.
@a bedeutet, dass die openFT-Script-Optionen aller aktiven openFT-Script-Benutzer sowie aller openFT-Script-Benutzer, die ein vom Standard abweichendes openFT-Script-Arbeitsverzeichnis haben, angezeigt werden.

Ausgabe im Tabellenformat

User	FtscriptWorkdir
<Benutzer>	<Pfadname>

<Benutzer>
Benutzerkennung

<Pfadname>

bezeichnet den Namen des openFT-Script-Arbeitsverzeichnisses, das der Benutzer mit *ftmodsuo* eingestellt hat, ohne die von openFT-Script angelegten Unterverzeichnisnamen.

Falls der Benutzer kein spezielles Arbeitsverzeichnis eingestellt hat, wird der Name seines Home-Verzeichnisses ausgegeben, da dieses standardmäßig openFT-Script-Arbeitsverzeichnis ist und zur Ablage der openFT-Script-Aufträge verwendet wird.

Ausgabe im CSV-Format

Spalte	Typ	Werte
User	String	Benutzerkennung
FtscriptWorkdir	String	Name des openFT-Script-Arbeitsverzeichnisses

Returncode

- 0 OK
- 4 Syntaxfehler

3.6 ftscript - openFT-Script-Auftrag starten

Das Kommando *ftscript* überprüft die angegebene Script-Datei und führt die darin enthaltenen Anweisungen aus. Die Script-Datei muss ein gültiges XML-Dokument enthalten, das dem Schema der openFT-Script-Schnittstelle entspricht. Außerdem muss die Datei mit der Berechtigung des Aufrufers lesbar sein. Es können maximal 1024 Benutzer openFT-Script-Aufträge besitzen. Hierbei zählen auch terminierte, aber noch nicht gelöschte openFT-Script-Aufträge.

Gibt es bei der Überprüfung Fehler, dann wird die Script-Datei nicht gestartet und die Fehler werden auf *stderr* ausgegeben.

Bei erfolgreichem Start der Script-Datei erscheint auf *stderr* die Meldung:

```
ftscript: erfolgreich gestartet. Id : ftscript id
```

Informationen über den openFT-Script-Auftrag werden während des Ablaufes und danach bis zum Ablauf der Löschfrist im openFT-internen Benutzerspeicher gehalten, so dass der Anwender sich mittels der ausgegebenen *ftscript id* über den Stand sowie Erfolg der Bearbeitung informieren kann.

ftscript ist wiederanlauffähig, d.h. die Bearbeitung des openFT-Script-Auftrags ist selbst nach einem Systemausfall sichergestellt.

Format

```
ftscript -h |  
          [ -t ]  
          <Ftscriptdateiname>
```

Beschreibung

- h** gibt die Kommandosyntax am Bildschirm aus. Weitere Angaben nach *-h* werden ignoriert.
- t** Es wird Diagnoseinformation (Trace) erstellt.

Ftscriptdateiname

Name der Script-Datei, die die XML-Anweisungen für den auszuführenden openFT-Script-Auftrag enthält.

Returncode

0	OK
4	Syntaxfehler
50	Ftscript-Prozess konnte nicht gestartet werden
52	Maximale Anzahl der Ftscript-Benutzer (1024) überschritten
55	Ftscript-Datei nicht gefunden
250	Interner Fehler

3.7 ftshwact - Aktivitäten eines openFT-Script-Auftrags anzeigen

Gibt Informationen über die einzelnen Aktivitäten des angegebenen openFT-Script-Auftrags aus.

Format für Unix-Systeme

```
ftshwact -h |
  [ -csv ]
  [ -a=<Id der Aktivität> | -d=<Ebenentiefe 1...> | -c=<Kapitel> ]
  [ -st=[W][R][T][F][K][D][C] ]
  [ -u=<Benutzerkennung 1..32> ]
  <ftscriptid>
```

Format für Windows-Systeme

```
ftshwact -h |
  [ -csv ]
  [ -a=<Id der Aktivität> | -d=<Ebenentiefe 1...> | -c=<Kapitel> ]
  [ -st=[W][R][T][F][K][D][C] ]
  [ -u=<Benutzerkennung 1..36> ]
  <ftscriptid>
```

Beschreibung

- h** gibt die Kommandosyntax am Bildschirm aus. Weitere Angaben nach *-h* werden ignoriert.
- csv** Die Informationen werden im CSV-Format ausgegeben. Wenn Sie *-csv* nicht angeben, werden die Informationen im Tabellenformat ausgegeben.
- a=Id der Aktivität**
Es wird nur die angegebene Aktivität angezeigt.
Sie können sich auf eine einzelne Anweisung eines Auftrags beziehen.
Die Id der Aktivität können Sie mit einem vorhergehenden *ftshwact*-Kommando (ohne Option *-a*) ermitteln. So können Sie den Stand der Aktivität zu einem späteren Zeitpunkt erneut ansehen.

-d=Ebenentiefe

Tiefe (depth) der anzuzeigenden Ebenen.

Es werden alle Aktivitäten angezeigt, deren Aktivitäts-Ids nicht mehr als die angegebene Stufenanzahl haben. Die Stufenanzahl ist die Anzahl der Indexnummern, die von Punkten getrennt werden.

Beispiel:

von einem Auftrag mit den Aktivitäts-Ids 1, 1.2, 1.2(1).1, 1.2(1).2, 1.2(2).1, 1.2(2).2 und 1.3 selektiert die Option `-d=2` die Aktivitäten mit den Aktivitäts-Ids 1, 1.2 und 1.3.

-c=Kapitel

Kapitel (chapter) der anzuzeigenden Aktivitäten.

Es werden diejenigen Aktivitäten ausgegeben, die eine Stufe unter der Aktivität mit der als Kapitel angegebenen Aktivitäts-Id stehen.

Im oben angegebenen Beispiel sind dies bei `-c=1`: 1.2 und 1.3;
bei `-c=1.2`: 1.2(1).1, 1.2(1).2, 1.2(2).1 und 1.2(2).2.

-st=[W][R][T][F][K][D][C]

Aktivitäten mit dem angegebenen Status anzeigen.

Sie können mehrere Status hintereinander angeben, z.B. `-st=WRT`.

Die Aktivität 1 wird in jedem Fall ausgegeben, da sie den Ausführungsstand des gesamten Scripts anzeigt.

-u=Benutzerkennung

Benutzerkennung, unter der der angegebene Auftrag gesucht wird.

Die Angabe einer Benutzerkennung ist nur für den FT-Verwalter erlaubt.

Standard ist die Benutzerkennung des Aufrufers.

ftscriptid

Identifikation des openFT-Script-Auftrags. Diese wird ausgegeben, wenn der openFT-Script-Auftrag per `ftscript`-Kommando gestartet wird.

Sie müssen genau einen openFT-Script-Auftrag angeben. Wildcard-Syntax wird nicht unterstützt.

Returncode

0	OK
4	Syntaxfehler
51	Fehler beim Ausgeben eines Ftscript-Benutzers
53	Ftscript-Kapitel nicht gefunden
54	Ftscript Id nicht gefunden
250	Interner Fehler

Beschreibung der Ausgabe

Die Ausgabe ist im Tabellenformat und im CSV-Format möglich.

Für noch nicht gestartete Aktivitäten ist die Ausgabe des *ftshwact*-Kommandos im Allgemeinen unvollständig, da die im Auftrag enthaltenen Referenzen noch nicht aufgelöst sind. Daher können nicht alle vorgesehenen Ausgabewerte versorgt werden. Insbesondere stehen Datei- oder Verzeichnisnamen bei Angabe einer Referenz erst zur Laufzeit fest, da sie Betriebssystem-spezifisch sind.

Ausgabe im Tabellenformat

Der Bearbeitungsstand der Aktivitäten wird in vier Spalten angezeigt:

Id Eindeutige Kennung der Aktivität innerhalb des Auftrags. Sie kann in einen Xpath umgewandelt werden, der die Stellung der Aktivität im Baum widerspiegelt, der statisch durch das XML-Script vorgegeben ist.

Lediglich für die *foreach*-Knoten kommt eine dynamische Information (Laufnummer in der *foreach*-Sequenz) hinzu.

Näheres siehe Handbuch „openFT-Script-Schnittstelle“ mit der Beschreibung der XML-Anweisungen.

Sta Status der Anweisung. Folgende Statuskennzeichen sind möglich:

W (waiting) Die Aktivität wurde noch nicht begonnen.

R (running) Die Aktivität wurde gestartet, ist aber noch nicht beendet.

T (terminated) Die Aktivität wurde ohne Fehler beendet.

F (failure) Die Aktivität wurde mit Fehler beendet.

K (killed) Die Aktivität wurde durch einen *faulhandler* oder ein *ftcans*-Kommando abgebrochen.

D (dead) Die Aktivität läuft aufgrund eines vorangegangenen Fehlers nicht mehr an.

Für die Aktivität *ftscript* (erste Aktivität eines openFT-Script-Auftrags) werden zusätzlich folgende Status unterschieden:

I (interrupted) Der Auftrag wurde unterbrochen, z.B. aufgrund eines Systemabsturzes.

C (cancelled) Der Auftrag wurde mit *ftcans* abgebrochen.

X (cancelling) Der Auftrag wird derzeit aufgrund eines *ftcans*-Kommandos abgebrochen.

F (failure) Wird für die Aktivität *ftscript* nur angezeigt, wenn der Fehler nicht durch einen *faulthandler* behandelt wurde.

Für Aktivitäten im Status F sowie für die *faulthandler*-Aktivität wird in einer weiteren Zeile die Fehlerursache in Klartext ausgegeben.

Activity

Aktivitäts-Name. Die Namen sind an die Sprache von openFT-Script angelehnt, jedoch teilweise abgekürzt, z.B. *faulthdlr* statt *faulthandler*.

foreach wird je nach dem Wert des Attributs *execute* mit *foreach_P* (parallel) bzw. *foreach_S* (sequentiell) bezeichnet.

TransferFile wird je nach Übertragungsrichtung mit *sendFile* oder *rcvFile* (=receive File) bezeichnet.

ActivityObject

Der Inhalt dieser Spalte ist abhängig von der jeweiligen Aktivität, siehe folgende Tabelle.

Activity	ActivityObject	Bedeutung
ftscript	<scriptPath>	Vollständiger Pfadname der ursprünglichen Datei mit den XML-Anweisungen.
empty	-	
foreach_P	<contextObject>	Kontextobjekt, das den Wert des aktuellen Listenelements annimmt.
foreach_S	wie foreach_P	
parallel	-	
sequence	-	

Activity	ActivityObject	Bedeutung
sendFile	Angabe der entfernten Datei in folgender Form:	
	<partner>!<Dateiname>	Partner mit Dateinamen, wenn beides bekannt ist.
	*unknown!<Dateiname>	wenn der Partner noch nicht bekannt ist.
	*unknown!*unknown	wenn beides noch nicht bekannt ist.
	<partner>!*ref(<contextId>)	falls <i>contextId = foreach contextObject</i> und die Auflösung noch nicht bekannt ist, da noch nicht durchlaufen.
	<Dateiname>	ist bei bereits gestarteten Aufträgen der im openFT-Auftrag angegebene Name. Bei noch nicht gestarteten Aufträgen wird dieser Name aus dem in der XML-Datei angegebenen Betriebssystem-spezifischen Namen (z.B. unixName) hergeleitet und um directory-Angaben ergänzt.
rcvFile	wie sendFile.	
deleteFile	Angabe der entfernten Datei wie <i>sendFile</i> (mit Partner); wenn die Datei lokal ist, ohne Partner	
	<Dateiname>	wird, wie bei <i>sendFile</i> , bei bereits gestarteten Aufträgen aus dem openFT-Auftrag, bei noch nicht gestarteten Aufträgen aus den Angaben der XML-Datei ermittelt. Ein lokaler Dateiname würde bei einem bereits gestarteten Auftrag als absoluter Pfadname, bei noch nicht gestartetem Auftrag ggf. relativ ausgegeben.
	*unknown!<Dateiname>	falls nicht bekannt ist, ob die Datei lokal ist, wenn ein Dateiojekt referenziert wird.
createDir	<partner>!<Verzeichnisname>	Partner mit Verzeichnisnamen, wenn beides bekannt ist.
	*unknown!<Verzeichnisname>	wenn der Partner noch nicht bekannt ist.
	*unknown!*unknown	wenn beides noch nicht bekannt ist.
	<partner>!*ref(<contextID>)	falls <i>contextId = foreach contextObject</i> und die Auflösung noch nicht bekannt ist, da noch nicht durchlaufen.

Activity	ActivityObject	Bedeutung
createDir (Forts.)	<Verzeichnisname>	wenn das Verzeichnis lokal ist. In diesem Fall wird der Name wie bei <i>sendFile</i> bei bereits gestarteten Aufträgen aus dem openFT-Auftrag und bei noch nicht gestarteten Aufträgen aus den Angaben der XML-Datei ermittelt. Ein lokaler Dateiname würde bei einem bereits gestarteten Auftrag als absoluter Pfadname, bei noch nicht gestartetem Auftrag ggf. relativ ausgegeben.
deleteDir	wie bei createDir.	
listDir	wie bei createDir.	
execScript	32 Zeichen	Enthält die ersten 32 Zeichen des auszuführenden Kommandos. Der Anwender sollte aus Sicherheitsgründen dafür sorgen, dass in den ersten 32 Zeichen keine geheimen Parameter enthalten sind.
fault	<faultcode>	Fehlercode, den der Anwender wirft.
faulthdl	<auslösende activity id>: <spezieller faultcode>; <allgemeiner faultcode>	

Ausgabe im CSV-Format

Die Ausgabe enthält folgende Informationen:

Id	Siehe Tabellen-Format auf Seite 42 .
State	Siehe Tabellen-Format auf Seite 42
Activity	Siehe Tabellen-Format auf Seite 43
ActivityObject	Siehe Tabellen-Format auf Seite 43 , eingeschlossen in Anführungszeichen, außer: <ul style="list-style-type: none"> – der Pfadname wird ohne Partner-Angabe ausgegeben – für die <i>faulthandler</i>-Aktivität werden nur die <i>faultcodes</i> ausgegeben.
Partner	Bei pfad-bezogenen Aktivitäten der Partner oder die Partner-Angabe, die im Tabellen-Format vor dem Pfadnamen stehen würde, eingeschlossen in Anführungszeichen. Sonst leer.
AddInfo	Bei <i>sendFile</i> und <i>rcvFile</i> : TID, eingeschlossen in Anführungszeichen, falls die Aktivität schon gestartet ist. Sonst leer. Bei <i>faulthdl</i> auslösende <i>activity-Id</i> , eingeschlossen in Anführungszeichen. Sonst leer.
nrElements	Bei gestartetem <i>foreach</i> : Anzahl der Schleifendurchläufe Bei gestartetem <i>parallel</i> oder <i>sequence</i> : Anzahl der Sub-Aktivitäten.
StartTime	Startzeit im Format yyyy-mm-dd hh:mm:ss
Error	Fehlerursache im Klartext bei Aufträgen im Status F, eingeschlossen in Anführungszeichen. Sonst leer.

3.8 ftshws - openFT-Script-Auftrag anzeigen

Gibt Informationen über den Status der openFT-Script-Aufträge eines Benutzers aus. Durch Angabe einer *ftscriptid* kann auch ein bestimmter openFT-Script-Auftrag ausgewählt werden.

Format für Unix-Systeme

```
ftshws -h |
        [ -csv ]
        [ -t ]
        [ -v ]
        [ -st=[W][R][T][F][I][C][X] ]
        [ -u=<Benutzerkennung 1..32> | @a ]
        [ <ftscriptid> ]
```

Format für Windows-Systeme

```
ftshws -h |
        [ -csv ]
        [ -t ]
        [ -v ]
        [ -st=[W][R][T][F][I][C][X] ]
        [ -u=<Benutzerkennung 1..36> | @a ]
        [ <ftscriptid> ]
```

Beschreibung

- h** gibt die Kommandosyntax am Bildschirm aus. Weitere Angaben nach *-h* werden ignoriert.
- csv** Die Informationen werden im CSV-Format ausgegeben. Wenn Sie *-csv* nicht angeben, werden die Informationen im Tabellenformat ausgegeben.
- t** Die openFT-Script-Aufträge werden nach Erzeugungszeitpunkt sortiert angezeigt, beginnend mit dem letzten Auftrag.

Standardmäßig werden die Aufträge in alphabetischer Reihenfolge angezeigt.

-v Es wird zusätzlich diagnoserelevante Information ausgegeben (verbose).

Ist `-v` angegeben, wird bei fehlerhaft beendeten openFT-Script-Aufträgen die Fehlerursache im Klartext in einer zweiten Zeile hinter der tabellarischen Information ausgegeben.

Im CSV-Format wird die Option `-v` ignoriert.

--st=[W][R][T][F][I][C][X]

zeigt openFT-Script-Aufträge mit dem angegebenen Status an, siehe Feld *Sta* auf [Seite 49](#).

Sie können mehrere Status hintereinander angeben, z.B. `-st=WRT`.

-u=Benutzerkennung | @a

Benutzerkennung, deren openFT-Script-Aufträge auszugeben sind bzw. unter der angegebenen Auftrag gesucht wird.

Die Angabe einer Benutzerkennung oder `@a` (alle Benutzerkennungen) ist nur für den FT-Verwalter erlaubt.

Standard ist die Benutzerkennung des Aufrufers.

ftscriptid

Identifikation des openFT-Script-Auftrags. Diese wird ausgegeben, wenn der openFT-Script-Auftrag per *ftscript*-Kommando gestartet wird.

In der *ftscriptid* können Sie die Wildcard-Symbole `?` und `*` verwenden. Dann werden alle openFT-Script-Aufträge ausgegeben, die dem Wildcard-Muster entsprechen.

`?` wird als genau ein beliebiges Zeichen interpretiert.

`*` wird als beliebig viele, beliebige Zeichen interpretiert.

Geben Sie bei Wildcard-Verwendung die *ftscriptid* in Hochkommas eingeschlossen an, damit die Wildcard-Symbole nicht durch die Shell interpretiert werden.

Standardmäßig, wenn Sie keine *ftscriptid* angeben, werden alle openFT-Script-Aufträge des Benutzers angezeigt.

Returncode

0	OK
4	Syntaxfehler
51	Fehler beim Ausgeben eines Ftscript-Benutzers
54	Ftscript Id nicht gefunden
250	Interner Fehler

Ausgabe im Tabellen-Format

Der Bearbeitungsstand der openFT-Script-Aufträge wird in vier Spalten angezeigt:

User Benutzerkennung, unter der der Auftrag gestartet wurde.

Ftscriptid

Eindeutige Identifikation des Auftrags. Die Identifikation wird vom *ftscript*-Kommando zurückgegeben.

Sta Zeigt den Bearbeitungsstatus an, dabei bedeuten:

W (waiting) Der Auftrag wurde noch nicht begonnen.

R (running) Der Auftrag wurde gestartet, ist aber noch nicht beendet.

T (terminated) Der Auftrag wurde ohne Fehler beendet.

F (failure) Der Auftrag wurde mit Fehler beendet.

I (interrupted) Der Auftrag wurde unterbrochen, z.B. aufgrund eines Systemabsturzes.

C (cancelled) Der Auftrag wurde mit einem *ftcans*-Kommando abgebrochen.

X (cancelling) Der Auftrag wird derzeit aufgrund eines *ftcans*-Kommandos abgebrochen.

FtscriptFileName

Pfadname der Script-Datei.

Falls der Status F und die Option *-v* angegeben ist, wird in einer weiteren Spalte die Fehlerursache im Klartext ausgegeben.

Ausgabe im CSV-Format

User;Ftscriptid;State;CreationTime;FtscriptFileName;Error;

Die Ausgabe enthält folgende Informationen:

User	Benutzerkennung, unter der der Auftrag gestartet wurde.
Ftscriptid	Eindeutige Identifikation des Auftrags. Die Identifikation wird vom <i>ftscript</i> -Kommando zurückgegeben.
State	Siehe Tabellen-Format (Sta).
CreationTime	Zeit, zu der der openFT-Script-Auftrag angelegt wurde, im Format yyyy-mm-dd hh:mm:ss.
FtscriptFileName	Pfadname der Script-Datei.
Error	Fehlerursache im Klartext bei openFT-Script-Aufträgen im Status F. Sonst leer.

User, *Ftscriptid*, *FtscriptFileName* und ggf. *Error* werden eingeschlossen in Anführungszeichen ausgegeben.

4 openFT-Script-Anweisungen

Das folgende Kapitel beschreibt die einzelnen openFT-Script-Anweisungen in alphabetischer Reihenfolge:

- Die Verwendung der Anweisung.
- Die vorhandenen Einschränkungen.
- Das Format beschreibt die Syntax der Anweisung. Zur Erläuterung der Syntax siehe folgenden [Abschnitt „Syntax der openFT-Script-Anweisungen“](#).
- Die vorhandenen Attribute der Anweisung, deren Werte und die Bedeutung der Werte. Optionale Attribute sind durch ein „?“ gekennzeichnet.
- Die Beispiele veranschaulichen die Anwendung der Anweisung.

4.1 Syntax der openFT-Script-Anweisungen

Die openFT-Script-Anweisungen sind in den folgenden Abschnitten beschrieben. Die Syntax der openFT-Script-Anweisungen ist jeweils im Abschnitt „Format“ angegeben. Dabei werden folgende Symbole verwendet:

Symbol	Bedeutung
a?	Kein Element oder ein Element a. Optionale Attribute sind ebenfalls durch ein „?“ gekennzeichnet.
a*	Kein Element oder beliebig viele Elemente a.
a+	Ein Element a oder mehrere Elemente a.
 a b+	Entweder ... oder ... Entweder (genau) ein Element a oder ein oder mehrere Elemente b.
< . . . >	In spitzen Klammern steht das aktuelle Element mit seinen Attributen. Elemente können zu einer Aktivität zusammengefasst werden. Solche zusammengefassten Elemente werden ohne spitze Klammern geschrieben.

4.2 baseDir

Mit *baseDir* können Sie ein Basisverzeichnis für die folgenden openFT-Script-Anweisungen festlegen:

- *createDirectory*
Unter dem Basisverzeichnis wird das Verzeichnis angelegt, das Sie mit *createDirectory* erzeugen (siehe [Abschnitt „createDirectory“ auf Seite 55](#)).
- *deleteDirectory*
Ausgehend von dem Basisverzeichnis wird das Verzeichnis gelöscht, das Sie mit *deleteDirectory* löschen (siehe [Abschnitt „deleteDirectory“ auf Seite 57](#)).
- *deleteFile*
Ausgehend von dem Basisverzeichnis wird die Datei gelöscht, die Sie mit *deleteFile* löschen (siehe [Abschnitt „deleteFile“ auf Seite 59](#)).
- *listDirectory*
Ausgehend von dem Basisverzeichnis werden die Dateien/Verzeichnisse aufgelistet. Das Basisverzeichnis selber wird nicht als Ergebnis aufgelistet (siehe [Abschnitt „listDirectory“ auf Seite 80](#)).

baseDir können Sie nur in Zusammenhang mit den oben aufgelisteten openFT-Script-Anweisungen verwenden.

Format

Siehe entsprechende openFT-Script-Anweisung.

Attribute

Name	Wert	Bedeutung
dirnames	siehe Abschnitt „Verzeichnisnamen-Attribute“ auf Seite 21 .	

Beispiele

- Siehe [Abschnitt „createDirectory“ auf Seite 55](#).
- Siehe [Abschnitt „deleteDirectory“ auf Seite 57](#).
- Siehe [Abschnitt „listDirectory“ auf Seite 80](#).

4.3 comment

Mit *comment* verfassen Sie einen Kommentartext zu dem entsprechenden Element.

Mit Hilfe der Kommentare können Sie die Scripte beschreiben.

Mit XML-Kommentaren (<!--...-->) kommentieren Sie Script-intern.

Format

```
<comment>  
  text  
</comment>
```

4.4 context

Mit *context* definieren Sie Kontextobjekte und *faulthandler* für eine Aktivität.

Kontextobjekte sind Elemente, die in derselben oder in darunterliegenden Aktivitäten verwendet werden können, indem sie mit dem Attribut *ref* bzw. *listRef* referenziert werden. Sie können vom Typ *autoDataSpec*, *directory*, *file*, *list*, *partner* oder *script* sein

Ein Kontext existiert für jede Aktivität. Ist kein Kontextelement angegeben, wird implizit ein leerer Kontext angelegt.

Es sind alle Kontextobjekte und *faulthandler* der darüberliegenden Kontexte sichtbar, solange sie nicht durch Kontextobjekte oder *faulthandler* mit gleichem Namen verdeckt werden.

Format

```
<context>
  KontextObjekt*
  faulthandler?
</context>
```

4.5 createDirectory

Mit *createDirectory* erzeugen Sie ein Verzeichnis. Wenn Sie keinen Partner angeben, wird das Verzeichnis unter der lokalen Benutzerkennung angelegt.

Mit *baseDir* (siehe [Abschnitt „baseDir“ auf Seite 52](#)) können Sie ein Basisverzeichnis festlegen, unter dem das angegebene Verzeichnis angelegt wird.

Die Länge des Verzeichnisnamens (Länge von *baseDir* plus Länge von *dirnames*) ist begrenzt und abhängig von der openFT-Version. Die Länge ergibt sich aus der Anzahl der Zeichen plus 1 Zeichen, wenn *baseDir* nicht mit einem „/“ endet.



Beachten Sie das Verhalten beim Wiederanlauf (siehe [Abschnitt „Wiederanlauf“ auf Seite 25](#)).

Einschränkungen

Wenn Sie im Namens-Attribut einen Verzeichnispfad angeben, müssen alle Verzeichnisse bis auf die unterste Stufe bereits existieren.

Format

```
<createDirectory ref?="ID" faultIfExists?="yes|no" dirnames >
  comment?
  context?
  partner?
  baseDir?
</createDirectory>
```

Attribute

Name	Wert	Bedeutung
ref?	string	Referenz auf ein Verzeichnis-Kontextobjekt
faultIfExists?	yes <u>no</u>	Der Standardwert ist <i>no</i> . Existiert das Verzeichnis, wird <i>createDirectory</i> ohne Fehler beendet. Bei Angabe von <i>yes</i> wird <i>createDirectory</i> mit dem Fehlercode <i>ft_exists</i> abgebrochen, wenn das Verzeichnis existiert. Beim Wiederanlauf (siehe Abschnitt „Wiederanlauf“ auf Seite 25) wird, bei existierendem Verzeichnis, <i>createDirectory</i> mit dem Fehlercode <i>ft_recoveryCreateDirectory</i> abgebrochen. Dies kann auch beim Umschalten der Instanz erfolgen.
dirnames	siehe Abschnitt „Verzeichnisnamen-Attribute“ auf Seite 21 .	

Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <context>
    <directory id="myDir" name="myTmp" />
  </context>
  <createDirectory ref="myDir">
    <partner name="UnixP_1">
      <transferAdmission>
        <ftacAdmission ftacAdmission="FTACADM1" />
      </transferAdmission>
    </partner>
    <baseDir name="frg_eis_03" />
  </createDirectory>
</ftscript>
```

Erzeugt im Verzeichnis *frg_eis_03* das Verzeichnis *myTmp* für die FTAC-Zugangsberechtigung *FTACADM1* auf dem Rechner *UnixP_1*.

createDirectory wird ohne Fehler beendet, wenn das Verzeichnis schon vorhanden ist. Wenn *createDirectory* gelaufen ist, existiert das Verzeichnis.

Als *baseDir* ist *frg_eis_03* angegeben. Das zu erzeugende Verzeichnis wird referenziert. Der gesamte Verzeichnispfad (*baseDir* + *name*) ist *frg_eis_03/myTmp* mit 16 Zeichen.

Kann das gewünschte Verzeichnis *frg_eis_03/myTmp* nicht angelegt werden, weil z.B. der Pfad *frg_eis_03* nicht existiert, wird *createDirectory* mit dem Fehlercode *ft_cantCreate* beendet.

Ein weiteres Beispiel für *createDirectory* finden Sie im [Abschnitt „foreach“ auf Seite 74](#).

4.6 deleteDirectory

Mit *deleteDirectory* löschen Sie ein Verzeichnis. Wenn Sie keinen Partner angeben, wird das Verzeichnis unter der lokalen Benutzerkennung gelöscht.

Mit *baseDir* können Sie ein Basisverzeichnis angeben (siehe [Abschnitt „directory“ auf Seite 61](#)), unter dem das zu löschende Verzeichnis gesucht wird. Der Basisverzeichnisname und der mit dem Attribut *dirnames* angegebene Name werden zum Verzeichnisnamen zusammengesetzt.

Die Länge des Verzeichnisnamens (Länge von *baseDir* plus Länge von *dirnames*) ist begrenzt und abhängig von der openFT-Version. Die Länge ergibt sich aus der Anzahl der Zeichen plus 1 Zeichen, wenn *baseDir* nicht mit einem „/“ endet.



Beachten Sie das Verhalten beim Wiederanlauf (siehe [Abschnitt „Wiederanlauf“ auf Seite 25](#)).

Einschränkungen

Das zu löschende Verzeichnis muss leer sein.

Format

```
<deleteDirectory ref?="ID" faultIfExists?="yes|no" dirnames>
  comment?
  context?
  partner?
  baseDir?
</deleteDirectory>
```

Attribute

Name	Wert	Bedeutung
ref?	string	Referenz auf ein Verzeichnis-Kontextobjekt
faultIfExists?	yes no	Der Standardwert ist <i>no</i> . Wenn das Verzeichnis nicht existiert, wird <i>deleteDirectory</i> ohne Fehler beendet. Bei Angabe von <i>yes</i> wird <i>deleteDirectory</i> mit dem Fehlercode <i>ft_notexists</i> beendet, wenn das Verzeichnis nicht existiert. Beim Wiederanlauf (siehe Abschnitt „Wiederanlauf“ auf Seite 25) wird <i>deleteDirectory</i> mit dem Fehlercode <i>ft_recoveryFailed</i> beendet. Dies kann auch beim Umschalten der Instanz erfolgen.
dirnames	siehe Abschnitt „Verzeichnisnamen-Attribute“ auf Seite 21 .	

Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <context>
    <directory id="myDir" name="myTmp" />
  </context>
  <deleteDirectory ref="myDir">
    <partner name="UnixP_1" systemType="unix">
      <transferAdmission>
        <ftacAdmission ftacAdmission="FTACADM1" />
      </transferAdmission>
    </partner>
    <baseDir name="frg_eis_04" />
  </deleteDirectory>
</ftscript>
```

Löscht das Verzeichnis *myTmp* für die FTAC-Zugangsberechtigung *FTACADM1* auf dem Rechner *UnixP_1*.

deleteDirectory wird ohne Fehler beendet, wenn das Verzeichnis nicht existiert oder schon gelöscht wurde.

Als *baseDir* ist *frg_eis_04* angegeben. Das zu löschende Verzeichnis wird referenziert. Der gesamte Verzeichnispfad (*baseDir* + *name*) ist *frg_eis_04/myTmp* mit 16 Zeichen.

4.7 deleteFile

Mit *deleteFile* löschen Sie eine Datei. Wenn Sie keinen Partner angeben, wird die Datei unter der lokalen Benutzerkennung gelöscht.

Der Name der zu löschenden Datei setzt sich aus dem mit *directory* angegebenen Verzeichnisnamen (siehe [Abschnitt „directory“ auf Seite 61](#)) und dem mit dem Attribut *filenames* angegebenen Namen zusammen.

Wenn Sie alle Dateien in einem Verzeichnis löschen wollen, verwenden Sie *listDirectory* (siehe [Abschnitt „listDirectory“ auf Seite 80](#)) in Kombination mit *foreach* (siehe [Abschnitt „foreach“ auf Seite 74](#)).

Die Länge des Dateinamens (Länge von *directory* plus Länge von *filenames*) ist begrenzt und abhängig von der openFT-Version. Die Länge ergibt sich aus der Anzahl der Zeichen plus 1 Zeichen, wenn *directory* nicht mit einem „/“ endet.



Beachten Sie das Verhalten beim Wiederanlauf (siehe [Abschnitt „Wiederanlauf“ auf Seite 25](#)).

Format

```
<deleteFile ref?="ID" faultIfNotExists?="yes|no" filenames >
  comment?
  context?
  partner?
  directory?
</deleteFile>
```

Attribute

Name	Wert	Bedeutung
ref?	string	Referenz auf ein Datei-Kontextobjekt
faultIfNotExists?	yes no	Der Standardwert ist <i>no</i> . Wenn die Datei nicht existiert, gilt dies nicht als Fehler; die Ftscript-Bearbeitung wird fortgesetzt. Bei Angabe von <i>yes</i> wird der Ftscript-Auftrag mit <i>ft_notExists</i> abgebrochen, wenn die Datei nicht existiert. Beim Wiederanlauf (siehe Abschnitt „Wiederanlauf“ auf Seite 25) wird der Ftscript-Auftrag mit <i>ft_recoveryFailed</i> abgebrochen. Dies kann auch beim Umschalten der Instanz erfolgen.
filenames	siehe Abschnitt „Dateinamen-Attribute“ auf Seite 20 .	

Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <deleteFile name="hugo.trash">
    <partner name="UnixP_1" systemType="unix">
      <transferAdmission>
        <ftacAdmission ftacAdmission="FTACADM1" />
      </transferAdmission>
    </partner>
    <directory name="frg_eis_05" />
  </deleteFile>
</ftscript>
```

Löscht die Datei *hugo.trash* im Verzeichnis *frg_eis_05* für die FTAC-Zugangsberechtigung *FTACADM1* auf dem Rechner *UnixP_1*.

Fehler werden in diesem Beispiel nicht behandelt und führen zum Abbruch des Ftscripts.

4.8 directory

Mit *directory* definieren Sie einen Verzeichnispfad.

directory ist immer ein Kind-Element einer Ftscript-Aktivität. Wenn ein fernes Verzeichnis gemeint ist, spezifizieren Sie den Partner als weiteres Unterelement der Aktivität.

Format

```
<directory id="ID" ref?="ref" dirnames >  
  comment?  
</directory>
```

Attribute

Name	Wert	Bedeutung
id	string	Eine eindeutige Id im aktuellen Kontext, unter der das Kontextobjekt referenziert wird.
ref?	string	Name des Verzeichnis-Kontextobjekts. Daten, die hier nicht vorhanden sind, werden von dem Verzeichnis-Kontextobjekt nach der Dereferenzierung übernommen.
dirnames	siehe Abschnitt „Verzeichnisnamen-Attribute“ auf Seite 21 .	

4.9 empty

Die Aktivität *empty* bewirkt nichts, wird jedoch formal benötigt, um in einem *faulthandler* einen Fehler ohne weitere Aktionen abzufangen (siehe [Abschnitt „faulthandler“ auf Seite 69](#)).

Format

```
<empty>
  comment?
</empty>
```

Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <transferFile>
    <context>
      <faulthandler>
        <default>
          <empty/>
        </default>
      </faulthandler>
    </context>
    <fromRemoteFile name="pack1.bin">
      <partner name="someHost">
        <transferAdmission>
          <ftacAdmission ftacAdmission="FTACADM1" />
        </transferAdmission>
      </partner>
      <directory name="frg_eis_06" />
    </fromRemoteFile>
    <toLocalFile name="target.bin">
      <directory name="frg_eis_06" />
    </toLocalFile>
  </transferFile>
</ftscript>
```

Wenn bei *transferFile* Fehler auftreten, wird der Ablauf des Ftscripts trotzdem weitergeführt, d.h. der openFT-Script-Auftrag wird mit Status T beendet.

Bei „schweren“ Fehlern (siehe [Abschnitt „Schwere“ Ftscript-Fehlercodes“ auf Seite 24](#)) wird der openFT-Script-Auftrag mit dem entsprechenden Fehlercode beendet, da der (siehe [Abschnitt „faulthandler“ auf Seite 69](#)) *default faulthandler* nicht wirkt.

4.10 executeScript

Mit *executeScript* führen Sie ein Script aus.

Das Script wird im Zielsystem ausgeführt. Wenn Sie keinen Partner angeben, wird das Script auf dem lokalen System unter der Benutzerkennung des Ftscrip-Aufrufers ausgeführt.

Folgende Kommando-Interpreter werden verwendet:

Betriebssystem	Kommando-Interpreter
Windows	Systemaufruf, d.h. es wird eine ausführbare Datei mit dem angegebenen Namen gesucht. Um z.B. ein Shell-Kommando auszuführen, ist <i>cmd /c</i> einzugeben.
Unix-System	<i>/bin/sh -c</i>
z/OS	TSO
BS2000	SDF

Einschränkungen

1. Ein Script (z.B. *bs2000Script*, *unixScript*) ist jeweils auf eine Zeile (ein Kommando) und in der Länge auf 500 Zeichen beschränkt.
2. Es muss für das angesprochene Betriebssystem ein nicht-leeres Script angegeben sein.

Format

```
<executeScript ref?="ID" repeatable?="true|false" >
  comment?
  context?
  script?
  bs2000Script?
  unixScript?
  windowsScript?
  zosScript?
  partner?
</executeScript>
```

Attribute

Name	Wert	Bedeutung
ref?	string	Referenz auf ein Script-Kontextobjekt
repeatable?	<u>true</u> false	<p>Der Standardwert ist <i>true</i>. Das Script darf beim Wiederanlauf wiederholt werden.</p> <p>Bei Angabe von <i>false</i>: Das Script darf beim Wiederanlauf nicht wiederholt werden. Bei einem Wiederanlauf wird die Aktivität <i>executeScript</i> mit dem Fehlercode <i>ft_resumeForbidden</i> abgebrochen, wenn nicht feststellbar ist, ob dieses Script durchgelaufen ist. Eine Umschaltung der openFT-Instanz während des Script-Ab- laufs wird nur für kurzlaufende Scripts unterstützt (siehe Abschnitt „Wiederanlauf“ auf Seite 25).</p>

Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <context>
    <list id="partnerList">
      <partner name="WindowsP_1" systemType="windows">
        <transferAdmission>
          <ftacAdmission ftacAdmission="FTACADM1" />
        </transferAdmission>
      </partner>
      <partner name="UnixP_1" systemType="unix">
        <transferAdmission>
          <ftacAdmission ftacAdmission="FTACADM2" />
        </transferAdmission>
      </partner>
    </list>
  </context>
  <foreach listRef="partnerList" selectType="partner"
    contextObject="partner">
    <executeScript>
      <unixScript><![CDATA[echo hello unix >frg_eis_07/demo.txt]]>
      </unixScript>
      <windowsScript><![CDATA[cmd /c echo 'hello windows'
        >frg_eis_07\demo.txt]]>
      </windowsScript>
      <partner ref="partner" />
    </executeScript>
  </foreach>
</ftscript>
```

Hier wird auf den Rechnern der Liste *partnerList* ein *executeScript* ausgeführt.

Je nach Betriebssystem werden die entsprechenden betriebssystemspezifischen Scripts ausgeführt. Wichtig ist die Angabe des Betriebssystems bei der Partnerdefinition.

4.11 fault

Mit *fault* brechen Sie die Eltern-Aktivität (und alle ihre laufenden Kind-Aktivitäten) mit einem selbst definierten Fehlercode ab und setzen die Ausführung im passenden *faulthandler* fort. Der *faulthandler* der Eltern-Aktivität wird zuerst durchlaufen.

Der Fehler wird mit dem passenden *faulthandler* (*default* oder *case*) abgefangen (siehe [Abschnitt „faulthandler“ auf Seite 69](#)). Die Aktivität des *faulthandler* wird ausgeführt.

Alle noch laufenden Aufträge zur Dateiübertragung, welche von der Eltern-Aktivität aus gestartet wurden, werden ebenfalls abgebrochen (*ft_cancel*). Dies führt gegebenenfalls zur Ausführung des *remoteFailureScript* (siehe [Abschnitt „remoteFailureScript“ auf Seite 109](#)).

Kann dem Fehlercode kein *faulthandler* zugeordnet werden, wird das gesamte Script abgebrochen.

Einschränkungen

Der Fehlercode darf nicht mit "ft_" beginnen. Diese Fehlercodes sind für openFT-Script reserviert.

Format

```
<fault code="faultcode">  
  comment?  
</fault>
```

Attribute

Name	Wert	Bedeutung
code	Text	Der Fehlercode, der in einem <i>faulthandler</i> abgefangen werden kann.

Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <parallel>
    <context>
      <partner id="remote" name="UnixP_1">
        <transferAdmission>
          <ftacAdmission ftacAdmission="FTACADM1" />
        </transferAdmission>
      </partner>
      <faulthandler>
        <case code="intercept">
          <deleteFile name="target1.bin">
            <partner ref="remote"/>
            <directory name="frg_eis_08NotAvailable"/>
          </deleteFile>
          <deleteFile name="target2.bin">
            <partner ref="remote"/>
            <directory name="frg_eis_08"/>
          </deleteFile>
        </case>
      </faulthandler>
    </context>
    <transferFile>
      <context>
        <faulthandler>
          <default>
            <fault code="intercept"/>
          </default>
        </faulthandler>
      </context>
      <fromLocalFile name
        "W:/openFT/ftscript/Test/data/small/bin.mp3"/>
      <toRemoteFile name="target1.bin">
        <partner ref="remote"/>
        <directory name="frg_eis_08NotAvailable"/>
      </toRemoteFile>
    </transferFile>
  </parallel>
</ftscript>
```

Beispiel (Forts.)

```
<transferFile>
  <context>
    <faulthandler>
      <default>
        <fault code="intercept"/>
      </default>
    </faulthandler>
  </context>
  <fromLocalFile name
    "W:/openFT/ftscript/Test/data/large/bin.mp3"/>
  <toRemoteFile name="target2.bin">
    <partner ref="remote"/>
    <directory name="frg_eis_08"/>
  </toRemoteFile>
</transferFile>
</parallel>
</ftscript>
```

Es werden zwei Dateiübertragungen parallel ausgeführt.

Tritt bei einer der Übertragungen ein Fehler auf, wird dieser auf den Fehlercode *intercept* abgebildet. Dieser wird im *faulthandler* abgefangen.

Die andere Dateiübertragung wird abgebrochen (*ft_cancel*).

Beide Zieldateien werden im *faulthandler* gelöscht. Hierbei auftretende Fehler werden ignoriert.

Mit diesem Script erreichen entweder beide Dateien oder keine der Dateien ihr Ziel.



Ist für die über *ft_cancel* abgebrochene Dateiübertragung ein *remoteFailureScript* definiert, kann dieses noch laufen, während der *faulthandler* bereits aktiv ist.

4.12 faulthandler

Der *faulthandler* wird analysiert, wenn in der ihn beinhaltenden Aktivität ein Fehler auftrat oder die Aktivität durch eine Kind-Aktivität in den Zustand „Fehler“ übergang (siehe [Abschnitt „Ausführung eines Ftscripts“ auf Seite 26](#)).

Der *faulthandler* wird angewendet, wenn in ihm eine *case* Aktivität mit dem entsprechenden Fehlercode definiert ist, oder, falls es sich nicht um einen „schweren“ Fehler handelt, eine *default* Aktivität (siehe [Abschnitt „Schwere“ Ftscript-Fehlercodes“ auf Seite 24](#)).

Diese *case* oder *default* Aktivität ersetzt dann die Aktivität, in der der *faulthandler* steht, übernimmt deren Kontextobjekte und wird ausgeführt.

Tritt beim Ablauf der *case* bzw. *default* Aktivität ein Fehler auf, so wird der ursprüngliche *faulthandler* nicht mehr berücksichtigt, sondern ein passender *faulthandler* in ihrem eigenen Kontext oder im Kontext ihrer Eltern-Aktivitäten.

Format

```
<faulthandler>  
  comment?  
  (case* default) | (case+ default?)  
</faulthandler>
```

Attribute

Siehe [Abschnitt „case“ auf Seite 71](#).

Beispiel

```

<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <context>
    <partner id="RemotePartner" name="D018S011">
      <transferAdmission>
        <ftacAdmission ftacAdmission="p11111111"/>
      </transferAdmission>
    </partner>
  </context>
  <transferFile>
    <context>
      <file id="source" name="source.bin"/>
      <file id="target" name="target.bin">
        <partner ref="RemotePartner"/>
      </file>
      <faulthandler>
        <default>
          <transferFile>
            <context>
              <partner id="RemotePartner" name="D018S022">
                <transferAdmission>
                  <ftacAdmission ftacAdmission="p22222222"/>
                </transferAdmission>
              </partner>
            </context>
            <fromLocalFile ref="source"/>
            <toRemoteFile ref="target"/>
          </transferFile>
        </default>
      </faulthandler>
    </context>
    <fromLocalFile ref="source"/>
    <toRemoteFile ref="target"/>
  </transferFile>
</ftscript>

```

In diesem Beispiel soll die lokale Datei *source.bin* nach *D018S011/target.bin* kopiert werden. Tritt beim *transferFile* ein impliziter Fehler auf, wird die Datei nach *D018S022/target.bin* kopiert. Das Kontextobjekt mit der Id *Remote Partner* wird dabei überdeckt. Tritt bei dieser Aktivität erneut ein Fehler auf, wird das Ftscript abgebrochen.

case

Die in *case* beschriebene Aktivität wird ausgeführt, wenn in Ihrer Liste von Fehlercodes der aktuelle Fehlercode gefunden wird. Die *case* Aktivität ersetzt die Aktivität, in der der *faulthandler* steht.

Die Kontextobjekte der ersetzten Aktivität werden in den Kontext der *case* Aktivität kopiert, sofern noch nicht vorhanden. Die Kontexte der Kind-Aktivitäten (auch die, in denen der Fehler aufgetreten ist) sind nicht mehr zugänglich.

Zum Ablauf siehe auch [Abschnitt „sequence“ auf Seite 91](#).

Format

```
<case code="codelist" >
  comment?
  context?
  Aktivität+
</case>
```

Attribute

Name	Wert	Bedeutung
code	codelist	Eine Liste von Fehlercodes, für die diese <i>case</i> Aktivität ausgeführt werden soll. Die einzelnen Fehlercodes sind durch Leerzeichen voneinander getrennt. Sie können hier „normale“ Ftscript-Fehlercodes (siehe Seite 23) und „schwere“ Ftscript-Fehlercodes (siehe Seite 24) verwenden, oder einen selbstdefinierten Fehlercode, den Sie mit der <i>fault</i> Aktivität erzeugen.

default

Wenn der aktuelle Fehlercode in keinem der *case* Elemente gefunden wird, wird die *default* Aktivität ausgeführt.

Dies gilt für alle Fehlercodes mit Ausnahme der „schweren“ Ftscript-Fehlercodes (siehe [Abschnitt „Schwere“ Ftscript-Fehlercodes“ auf Seite 24](#)), für die der *default* Zweig des *faulthandler* nicht berücksichtigt wird.

Die *default* Aktivität ersetzt die Aktivität, in der der *faulthandler* steht.

Die Kontextobjekte der ersetzten Aktivität werden in den Kontext der *default* Aktivität kopiert, sofern noch nicht vorhanden. Die Kontexte der Kind-Aktivitäten (auch die, in denen der Fehler aufgetreten ist) sind nicht mehr zugänglich.

Zum Ablauf siehe auch [Abschnitt „sequence“ auf Seite 91](#).

Format

```
<default>  
  comment?  
  context?  
  Aktivität*  
</default>
```

4.13 file

Mit *file* definieren Sie eine Datei.

Eigenschaften einer Datei:

- Die Datei hat einen System-spezifischen Namen.
- Die Datei liegt auf einem konkreten System (*partner*) in einem konkreten Verzeichnis (*directory*). Wenn kein Partner angegeben ist, liegt sie im lokalen System.

Einschränkungen

Es gelten die Einschränkungen für *filenames* und *directory* (siehe Abschnitte „Dateinamen-Attribute“ auf Seite 20 und „Verzeichnisnamen-Attribute“ auf Seite 21).

Format

```
<file id="ID" ref?="ref" filenames >
  comment?
  partner?
  directory?
</file>
```

Attribute

Name	Wert	Bedeutung
id	string	Eine eindeutige Id im aktuellen Kontext, unter der das Kontextobjekt referenziert wird.
ref?	string	Name eines anderen Datei-Kontextobjekts.
filenames	siehe Abschnitt „Dateinamen-Attribute“ auf Seite 20 .	

4.14 foreach

Mit *foreach* führen Sie eine Sequenz für jedes Element einer Liste aus (siehe [Abschnitt „list“ auf Seite 79](#)). Für jedes Element der selektierten Liste werden die Kind-Elemente von *foreach* als Sequenz ausgeführt.

Sie können festlegen, ob die Sequenzen nacheinander (in der Reihenfolge der Listenelemente) oder parallel ausgeführt werden.

Format

```
<foreach listRef="ID" contextObject="ID" execute?="parallel|sequential"
      selectType="file|partner|directory" direction?="forward|reverse" >
  comment?
  context?
  Aktivität+
</foreach
```

Attribute

Name	Wert	Bedeutung
listRef	string	Name eines gültigen Kontextobjekts vom Typ <i>list</i> .
contextObject	string	Name des <i>foreach</i> Kontextobjekts, das den Wert des aktuellen Listenelements annimmt. Dies darf im <i>foreach</i> Kontext nicht definiert sein. Es wird implizit definiert. Der Typ des Kontextobjekts entspricht dem im Attribut <i>selectType</i> eingestellten Typ.
execute?	parallel <u>sequential</u>	Der Standardwert ist <i>sequential</i> . Die Sequenzen werden nacheinander ausgeführt. Bei Angabe von <i>parallel</i> werden die Sequenzen parallel gestartet.
selectType	partner file directory	Filtert die Elemente des angegebenen Typs aus der Liste. Es wird nur über die gefilterten Elemente iteriert.
direction?	<u>forward</u> reverse	Der Standardwert ist <i>forward</i> . Die Liste wird vorwärts abgearbeitet. Bei Angabe von <i>reverse</i> wird die Liste rückwärts abgearbeitet.

Beispiele

1. Dateien verteilen

```

<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <context>
    <list id="FileList">
      <file name="bin.mp3"/>
      <file name="text.txt"/>
    </list>
    <list id="HostList">
      <partner name="UnixP_1" systemType="unix">
        <transferAdmission>
          <ftacAdmission ftacAdmission="FTACADM1"/>
        </transferAdmission>
      </partner>
      <partner name="WindowsP_1" systemType="windows">
        <transferAdmission>
          <ftacAdmission ftacAdmission="FTACADM2"/>
        </transferAdmission>
      </partner>
    </list>
  </context>
  <foreach listRef="HostList" selectType="partner"
    contextObject="partner" execute="parallel">
    <foreach listRef="FileList" selectType="file"
      contextObject="file" execute="parallel">
      <transferFile>
        <fromLocalFile ref="file">
          <directory name="W:/openFT/ftscript/Test/data/large"/>
          <autoDataSpec binPattern="*.mp3" charPattern="*.txt"/>
        </fromLocalFile>
        <toRemoteFile ref="file">
          <partner ref="partner"/>
          <directory name="frg_eis_09"/>
        </toRemoteFile>
      </transferFile>
    </foreach>
  </foreach>
</ftscript>

```

Es werden die Dateien *bin.mp3* und *text.txt* auf zwei Rechner kopiert.

Die Datei- und Rechnerlisten werden im Beispiel als Kontextobjekte definiert.

Die Dateiliste kann auch z.B. durch ein *listDirectory* definiert werden (siehe [Abschnitt „listDirectory“ auf Seite 80](#)).

Es wird eine doppelte *foreach* Sequenz verwendet. Die äußere Sequenz durchläuft alle Rechner, die innere Sequenz durchläuft alle Dateien. Die Verbindung zu den Rechnern findet parallel statt, die Dateien werden ebenfalls parallel zu jedem Rechner übertragen.

autoDataSpec unterscheidet zwischen Text- und Binärdateien (siehe [Abschnitt „autoDataSpec“ auf Seite 99](#)).

Wenn das Script ausgeführt ist, sind die Dateien auf alle Rechner verteilt. Da im Beispiel kein *faulthandler* verwendet wurde, bricht das Script bei einem Fehler ab.

2. Dateibaum kopieren

```
<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <context>
    <partner id="remote" name="UnixP_1">
      <transferAdmission>
        <ftacAdmission ftacAdmission="FTACADM1"/>
      </transferAdmission>
    </partner>
  </context>
  <listDirectory name="*//*" listObject="Flist">
    <partner ref="remote"/>
    <baseDir name="frg_eis_11"/>
  </listDirectory>
  <foreach listRef="Flist" selectType="directory"
    contextObject="creDir" execute="sequential">
    <createDirectory ref="creDir">
      <baseDir name="frg_eis_11"/>
    </createDirectory>
  </foreach>
  <foreach listRef="Flist" selectType="file"
    contextObject="file" execute="parallel">
    <transferFile>
      <fromRemoteFile ref="file">
        <partner ref="remote"/>
        <directory name="frg_eis_11"/>
      </fromRemoteFile>
      <toLocalFile ref="file">
        <directory name="frg_eis_11"/>
      </toLocalFile>
    </transferFile>
  </foreach>
</ftscript>
```

Im Verzeichnis *frg_eis_11* des Rechners *UnixP_1* wird der Dateibaum **//** in das Verzeichnis *frg_eis_11* der lokalen Kennung kopiert.

In der ersten *foreach* Sequenz werden alle (notwendigen) Verzeichnisse sequentiell mit *createDirectory* angelegt.

listDirectory liefert für das Verzeichnis *frg_eis_11/a/b/c* der Reihe nach die Verzeichnisse *a*, *a/b* und *a/b/c* (*frg_eis_11* selbst ist mit *baseDir* als Basisverzeichnis definiert). Ein Verzeichnis kann erst angelegt werden, wenn das Eltern-Verzeichnis existiert.

In der zweiten *foreach* Sequenz werden die Dateien parallel kopiert, da nun alle Zielverzeichnisse vorhanden sind.

3. Dateibaum löschen

```
<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <context>
    <partner id="remote" name="UnixP_1">
      <transferAdmission>
        <ftacAdmission ftacAdmission="FTACADM1"/>
      </transferAdmission>
    </partner>
  </context>
  <listDirectory name="frg_eis_10/*/*" listObject="Flist">
    <partner ref="remote"/>
  </listDirectory>
  <foreach listRef="Flist" selectType="file"
    contextObject="delFile" execute="parallel">
    <deleteFile ref="delFile">
      <partner ref="remote"/>
    </deleteFile>
  </foreach>
  <foreach listRef="Flist" selectType="directory"
    contextObject="delDir" execute="sequential"
    direction="reverse">
    <deleteDirectory ref="delDir">
      <partner ref="remote"/>
    </deleteDirectory>
  </foreach>
</ftscript>
```

Im Beispiel wird unter der FTAC-Zugangsberechtigung *FTACADM1* auf dem Rechner *UnixP_1* alles im Verzeichnis *frg_eis_10* gelöscht.

Mit *listDirectory* (siehe [Abschnitt „listDirectory“ auf Seite 80](#)) werden durch das Suchmuster **/** alle Dateien und Verzeichnisse rekursiv ermittelt. Die Reihenfolge in der die Verzeichnisse aufgelistet werden, entspricht der zu deren Erzeugung notwendigen Reihenfolge (d.h. die umgekehrte Reihenfolge ist zum Löschen notwendig).

In der ersten *foreach* Sequenz werden alle Dateien parallel gelöscht. Nicht existierende Dateien werden ignoriert. Fehler beim Löschen einer Datei führen zum Abbruch des Scripts.

In der zweiten *foreach* Sequenz werden die leeren Verzeichnisse rückwärts gelöscht, weil die Verzeichnisse zum Löschen mit *deleteDirectory* leer sein müssen (siehe [Abschnitt „deleteDirectory“ auf Seite 57](#)). Nicht existierende Verzeichnisse werden ignoriert. Andere Fehler führen zum Abbruch des Scripts.

Wenn das Script ausgeführt ist, ist das Verzeichnis *frg_eis_10* auf dem Rechner *UnixP_1* leer.

4.15 ftscript

ftscript ist das Wurzel-Element des Scripts.

Das Element entspricht einer *sequence* Aktivität (siehe [Abschnitt „sequence“ auf Seite 91](#)).

Format

```
<ftscript version="1">  
  comment?  
  context?  
  Aktivität+  
</ftscript>
```

Attribute

Name	Wert	Bedeutung
version	1	Fester Wert, der die Ftscript-Version beschreibt.

Beispiel

Siehe beliebiges Beispiel im vorliegenden Handbuch.

4.16 list

Eine Liste enthält mehrere Elemente vom Typ *partner*, *directory* oder *file*.

Eine Liste können Sie auch mit *listDirectory* erzeugen (siehe [Abschnitt „listDirectory“ auf Seite 80](#)).

Mit *foreach* können Sie über die Elemente der Liste iterieren (siehe [Abschnitt „foreach“ auf Seite 74](#)).

Format

```
<list id="ID" >  
  comment?  
  ( partner | directory | file )*  
</list>
```

Attribute

Name	Wert	Bedeutung
id	string	Eine eindeutige Id im aktuellen Kontext, unter der das Kontextobjekt referenziert wird.

Beispiel

Siehe [Abschnitt „foreach“ auf Seite 74](#).

4.17 listDirectory

Mit *listDirectory* listen Sie Dateien und Verzeichnisse auf. Die gefundenen Datei- oder Verzeichnisnamen werden in einer Liste zusammengefasst. Die Liste wird in einem Kontextobjekt mit der angegebenen *listObject-Id* in den Eltern-Kontext eingeblendet. Dort ist das Kontextobjekt nach dem Ausführen der *listDirectory* Aktivität verfügbar.

Tritt bei der Ausführung von *listDirectory* ein Fehler auf, ist das Objekt nicht verfügbar.



Ein *listDirectory* als direktes Kind-Element einer *parallel* Aktivität ergibt keine verwendbare Ergebnisliste (siehe [Abschnitt „parallel“ auf Seite 83](#)).

Mit *baseDir* (siehe [Abschnitt „baseDir“ auf Seite 52](#)) können Sie ein Basisverzeichnis festlegen, von dem aus *listDirectory* ausgeführt wird.

Das Basisverzeichnis selber wird nicht als Ergebnis aufgelistet.

Datei- oder Verzeichnisnamen dürfen maximal 512 Zeichen lang sein. Dies wird vor der Ausführung des Ftscript überprüft. Die Länge des resultierenden Datei- oder Verzeichnisnamens (*baseDir* und *dirnames* oder *baseDir* + *filenames*) ist durch das angesprochene Betriebssystem und die openFT-Version begrenzt. Die Länge ergibt sich aus der Anzahl der Zeichen plus 1 Zeichen, wenn *baseDir* nicht mit einem „/“ endet. Die Überprüfung findet während des Ftscript-Laufes statt.

Weitere Informationen dazu entnehmen Sie dem [Abschnitt „Datei- und Verzeichnisnamen angeben“ auf Seite 19](#).



Sie können das Wildcard-Symbol ***/** verwenden, um einen Verzeichnisbaum vollständig aufzulisten.

Format

```
<listDirectory listObject="ID" dirnames >
  comment?
  context?
  partner?
  baseDir?
</listDirectory>
```

Attribute

Name	Wert	Bedeutung
ref?	string	Referenz auf ein Verzeichnis-Kontextobjekt.
listObject	string	Name des <i>list</i> Kontextobjekts, welches in den Eltern-Kontext eingeblenet wird. Es darf kein Kontextelement mit diesem Namen existieren.
dirnames	siehe Abschnitt „Verzeichnisnamen-Attribute“ auf Seite 21 . Um einen Verzeichnisbaum vollständig aufzulisten, können Sie das Wildcard-Symbol <code>/**</code> am Ende des Namens verwenden.	

Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <context>
    <list id="RemoteHostList">
      <partner name="UnixP_1">
        <transferAdmission>
          <ftacAdmission ftacAdmission="FTACADM1"/>
        </transferAdmission>
      </partner>
      <partner name="WindowsP_1">
        <transferAdmission>
          <ftacAdmission ftacAdmission="FTACADM2"/>
        </transferAdmission>
      </partner>
    </list>
  </context>
  <listDirectory listObject="FileList">
    <baseDir name="W:/openFT/ftscript/Test/multi"/>
  </listDirectory>
  <foreach listRef="RemoteHostList" selectType="partner"
    contextObject="RemoteHost" execute="parallel">
    <foreach listRef="FileList" selectType="file"
      contextObject="File" execute="parallel">
      <transferFile>
        <fromLocalFile ref="File">
          <directory name="W:/openFT/ftscript/Test/multi"/>
        </fromLocalFile>
        <toRemoteFile ref="File">
          <partner ref="RemoteHost"/>
          <directory name="frg_eis_12/local/bin"/>
        </toRemoteFile>
      </transferFile>
    </foreach>
  </foreach>
</ftscript>
```

Im Beispiel werden aus dem lokalen Verzeichnis *W:/openFT/ftscript/Test/multi/* alle Dateien in das Kontextobjekt *FileList* geschrieben.

In der Liste stehen nur die Dateinamen, nicht das *baseDir*-Verzeichnis.

Die Dateien werden parallel auf alle Rechner der *RemoteHostList* nach *frg_eis_12/local/bin/* verteilt.

Das Kontextobjekt *FileList* aus der *listDirectory* Aktivität wird in den Kontext des Eltern-Elements (im Beispiel *ftscript*) eingeblendet.

4.18 parallel

Mit *parallel* führen Sie alle Aktivitäten „gleichzeitig“ und voneinander unabhängig aus. Eine „echte Gleichzeitigkeit“ (wie beispielsweise beim Zeitscheibenverfahren) ist jedoch nicht implementiert.

Die *parallel* Aktivität wird beendet, wenn alle Kind-Aktivitäten beendet sind. Wenn eine Kind-Aktivität einen Fehler ausgibt, werden noch laufende Kind-Aktivitäten abgebrochen.

Die Ergebnisse der einen Kind-Aktivität sind nicht in den anderen Kind-Aktivitäten sichtbar.

Eine Übergabe von Kontextobjekten in den Eltern-Kontext ist momentan nicht vorgesehen. Daher werden entstandene Kontextobjekte bei der *parallel* Aktivität verworfen.

i Ein *listDirectory* als direktes Kind-Element einer *parallel* Aktivität liefert daher keine verwendbare Ergebnisliste (siehe [Abschnitt „listDirectory“ auf Seite 80](#)).

parallel bezieht sich auf die parallele Bearbeitung der Aktivitäten durch openFT-Script. Die maximale Anzahl der tatsächlich parallel laufenden Dateiübertragungen wird durch das Verbindungslimit und das Prozesslimit von openFT beschränkt (siehe openFT-Handbuch „Installation und Administration“). Um Ressourcen zu schonen, stellt openFT-Script nur so viele Aufträge gleichzeitig ins Auftragsbuch, dass dieses Verbindungslimit nicht überschritten wird. Die Reihenfolge, in der die Aufträge abgearbeitet werden, ist bei *parallel* beliebig.

Auch synchrone Aktivitäten wie *deleteScript* werden im Rahmen einer *parallel* Aktivität nicht unbedingt alle gleichzeitig gestartet, sondern nur maximal 200, damit nicht zuviele Threads aufgebaut werden müssen.

i Die openFT-Betriebsparameter Verbindungslimit und Prozesslimit steuern die gleichzeitige oder sequenzielle Abarbeitung von Aufträgen aus dem Auftragsbuch, und beeinflussen damit die tatsächliche Parallelität sowie Performance und Ressourcenverbrauch Ihres openFT-Script-Auftrags.

Format

```
<parallel>
  comment?
  context?
  Aktivität+
</parallel>
```

Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <parallel>
    <transferFile>
      <fromLocalFile name=
        "W:/openFT/ftscript/Test/data/large/text.txt"
        data="char"/>
      <toRemoteFile name="text.txt">
        <partner name="UnixP_1">
          <transferAdmission>
            <ftacAdmission ftacAdmission="FTACADM1"/>
          </transferAdmission>
        </partner>
        <directory name="frg_eis_13"/>
      </toRemoteFile>
    </transferFile>
    <transferFile>
      <fromLocalFile name=
        "W:/openFT/ftscript/Test/data/large/bin.mp3"
        data="bin"/>
      <toRemoteFile name="bin.mp3">
        <partner name="WindowsP_1">
          <transferAdmission>
            <ftacAdmission ftacAdmission="FTACADM2"/>
          </transferAdmission>
        </partner>
        <directory name="frg_eis_13"/>
      </toRemoteFile>
    </transferFile>
  </parallel>
</ftscript>
```

Die Dateien *text.txt* und *bin.mp3* werden „gleichzeitig“ ausgeliefert.

Tatsächlich werden die Aufträge zur Dateiübertragung „gleichzeitig“ an openFT gesendet. Wenn genügend Kapazitäten frei sind, werden sie gleichzeitig ausgeführt.

4.19 partner

Mit *partner* spezifizieren Sie den Partner, für den die Aktivität gilt oder wo die Datei oder das Verzeichnis zu finden ist.

Wenn ein Partner angegeben ist, gilt dieser immer als fernes System, auch wenn der eigene Rechner und die eigene Benutzerkennung adressiert ist. Wenn Sie keinen Partner angeben, bezieht sich die Aktivität auf die aktuelle Benutzerkennung auf dem lokalen Rechner.

Nach der Dereferenzierung muss der Partner eine Übertragungsberechtigung (siehe [Abschnitt „transferAdmission“ auf Seite 87](#)) haben.

Format

```
<partner id="ID" ref?="ref" name="name" systemType?=
"any|unix|windows|zos|bs2000" >
  comment?
  transferAdmission
  processingAdmission?
</partner>
```

Attribute

Name	Wert	Bedeutung
id	string	Eindeutige Id im aktuellen Kontext, unter der dieses Kontextobjekt referenziert wird.
ref	string	Referenziert ein Kontextobjekt vom Typ Partner. Nach der Dereferenzierung muss der Partner eine Übertragungsberechtigung haben (siehe Abschnitt „transferAdmission“ auf Seite 87).
name	string	Name des Partnersystems. Zulässig sind TNS- und DNS-Namen. IPv4- und IPv6-Adressen beginnen mit %IP (siehe openFT-Benutzerhandbuch).
systemType?	any unix windows zos bs2000	Der Standardwert ist any. Gibt den Systemtyp des Partners an. Der Systemtyp wird nicht automatisch bestimmt. Wenn kein Systemtyp angegeben ist, werden anstelle der System-spezifischen Daten (z.B. unixname) die allgemeinen Daten (z.B. name) verwendet.

4.19.1 processingAdmission

Mit *processingAdmission* vergeben Sie die Ausführungsberechtigung für Scripts.



Momentan wird *processingAdmission* nicht unterstützt. Das Attribut kann angegeben werden, hat jedoch keine Auswirkungen.

Format

```
<processingAdmission userId?="user" userAccount?="account"
  userPassword?="passwort" >
  comment?
</processingAdmission>
```

Attribute

Name	Wert	Bedeutung
userId?	string	Benutzerkennung. Muss nicht angegeben werden, wenn die <i>transferAdmission</i> verwendet wird (Abschnitt „transferAdmission“ auf Seite 87).
userAccount?	string	Abrechnungsinformation.
userPassword?	string	Passwort der Benutzerkennung.

4.19.2 transferAdmission

Mit *transferAdmission* vergeben Sie die Berechtigung zur Dateiübertragung.

Format

```
<transferAdmission>
  comment?
  ftacAdmission | userAdmission
</transferAdmission>
```

ftacAdmission

Mit *ftacAdmission* geben Sie die Berechtigung in Form der FTAC-Zugangsberechtigung an.

Format

```
<ftacAdmission ftacAdmission="ftac" />
```

Attribute

Name	Wert	Bedeutung
ftacAdmission	string	FTAC-Zugangsberechtigung (siehe openFT-Benutzerhandbuch).

userAdmission

Mit *userAdmission* geben Sie die Berechtigung in Form der Login-/LOGON-Zugangsdaten an (Benutzerkennung, Passwort und Account).



Vermeiden Sie die *userAdmission* in dieser Version. openFT-Script erfordert die Angabe des Passworts im Klartext.

Wenn möglich, verwenden Sie stattdessen die sicherere *ftacAdmission*.

Format

```
<userAdmission userId="userId" userAccount?="account"  
  userPassword?="password" >  
  comment?  
</userAdmission>
```

Attribute

Name	Wert	Bedeutung
userId	string	Benutzerkennung.
userAccount?	string	Abrechnungsinformation.
userPassword?	string	Passwort der Benutzerkennung.

4.20 script

Mit *script* geben Sie einen Textstring an, der als Betriebssystem-Kommando ausgeführt werden soll. Ein Kontextobjekt vom Typ *script* kann in der *executeScript* Aktivität oder von *remoteSuccessScript* oder *remoteFailureScript* referenziert werden.

Das Kommando muss für jedes Betriebssystem in der betriebssystemspezifischen Syntax geschrieben sein. Sie können in einem *script* Objekt für jedes Betriebssystem einen anderen Textstring angeben.

openFT-Script wählt den auszuführenden Textstring gemäß der Angabe des Betriebssystems in der Partner-Definition oder gemäß dem lokalen Betriebssystem. Ist das Betriebssystem nicht bekannt oder kein zum Betriebssystem passendes *script* angegeben, wird der mit *script* bezeichnete Textstring ausgeführt.

Der Textstring (das *script*) wird beim *ftshwact*-Kommando vollständig ausgegeben. Wenn dieser String Passwörter oder ähnliche, sicherheitsrelevante Information enthält (z.B. bei einem *ncopy*-Kommando), dann werden diese mit dem *ftshwact*-Kommando angezeigt.

Ein Rückgabewert ungleich Null wird als Fehler interpretiert und führt zum *ft_scriptError*.

Einschränkungen

1. Ein Script (z.B. *bs2000Script*, *unixScript*) ist jeweils auf eine Zeile (ein Kommando) und in der Länge auf 500 Zeichen beschränkt.
2. In dieser Version ist das Script nicht parametrisierbar.

Format

```
<script id="ID" ref?="ID" repeatable?=true|false" >
  comment?
  script?
  bs2000Script?
  unixScript?
  windowsScript?
  zosScript?
  partner?
</script>
```

Attribute

Name	Wert	Bedeutung
id	string	Eindeutige Id im aktuellen Kontext, unter der dieses Kontextobjekt referenziert wird.
ref	string	Id eines anderen Script-Objekts.
repeatable?	<u>true</u> false	Der Standardwert ist <i>true</i> . Das Script darf beim Wiederanlauf wiederholt werden. Bei Angabe von <i>false</i> darf das Script nicht wiederholt werden. Kann beim Wiederanlauf nicht festgestellt werden, ob das Ftscript gelaufen ist, wird das gesamte Ftscript abgebrochen.

Beispiele

Siehe [Abschnitt „executeScript“](#) auf Seite 63.

4.21 sequence

Mit *sequence* führen Sie die Aktivitäten einer Sequenz nacheinander in angegebener Reihenfolge aus.

Jede Kind-Aktivität kann die Ergebnisse der voran gegangenen Kind-Aktivitäten nutzen. Wenn durch eine Aktivität dem *sequence* Kontext neue Kontextobjekte hinzugefügt werden (z.B. *transferFile/toLocalTmpFile* oder *listDirectory*), können die folgenden Aktivitäten auf die neuen Daten zugreifen.

Format

```
<sequence>
  comment?
  context?
  Aktivität+
</sequence>
```

Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <sequence>
    <transferFile>
      <fromLocalFile name=
        "W:/openFT/ftscript/Test/data/small/text.txt" data="char"/>
      <toRemoteFile name="text.txt">
        <partner name="UnixP_1">
          <transferAdmission>
            <ftacAdmission ftacAdmission="FTACADM1"/>
          </transferAdmission>
        </partner>
        <directory name="frg_eis_14"/>
      </toRemoteFile>
    </transferFile>
    <transferFile>
      <fromLocalFile name=
        "W:/openFT/ftscript/Test/data/large/bin.mp3" data="bin"/>
      <toRemoteFile name="bin.mp3">
        <partner name="WindowsP_1">
          <transferAdmission>
            <ftacAdmission ftacAdmission="FTACADM2"/>
          </transferAdmission>
        </partner>
        <directory name="frg_eis_14"/>
      </toRemoteFile>
    </transferFile>
  </sequence>
</ftscript>
```

Die Dateien *text.txt* und *bin.mp3* werden nacheinander ausgeliefert.

Zuerst wird der erste Auftrag in das Auftragsbuch gestellt. Wenn dieser abgeschlossen ist und keine Fehlermeldung ausgelöst wurde, wird der zweite Auftrag in das Auftragsbuch gestellt.

4.22 transferFile

Mit *transferFile* führen Sie eine Dateiübertragung aus. *transferFile* startet die Dateiübertragung und wartet innerhalb von *fiscript* auf das Ende der Dateiübertragung. Die Dateiübertragung selbst ist wiederanlauffähig.

Wenn Sie *remoteSuccessScript* (siehe [Abschnitt „remoteSuccessScript“ auf Seite 111](#)) oder *remoteFailureScript* (siehe [Abschnitt „remoteFailureScript“ auf Seite 109](#)) angeben, wird anschließend auf dem fernen Rechner das entsprechende Script ausgeführt. Für die lokalen Scripts sollten Sie *faulthandler* (siehe [Abschnitt „faulthandler“ auf Seite 69](#)) oder *executeScript* (siehe [Abschnitt „executeScript“ auf Seite 63](#)) verwenden (Beispiel siehe [Seite 98](#)).

Arten der Dateiübertragung

Mit *transferFile* können Sie Dateien folgendermaßen übertragen:

- Dateiübertragungen von „remote“ nach „local“ (*fromRemoteFile toLocalFile*)
- Dateiübertragungen von „local“ nach „remote“ (*fromLocalFile toRemoteFile*)
- Dateiübertragungen von „remote“ nach „remote“ sind über zwei aufeinander folgende (sequenzielle) *transferFile* Aktivitäten zu realisieren, z.B. in einer *sequence* Aktivität:
 - Dateiübertragung von „remote“ nach „localTmp“ (*fromRemoteFile toLocalTmpFile*)
 - Dateiübertragung von „localTmp“ nach „remote“ (*fromLocalTmpFile toRemoteFile*)

Einschränkungen

1. Nach der Dereferenzierung müssen *fromRemoteFile* und *toRemoteFile* eine Partnerangabe besitzen.
2. Die Elemente *fromLocalFile* und *toLocalFile* dürfen nach der Dereferenzierung keine Partnerangabe besitzen.
3. Mit *transferFile* können Sie keine Dateiübertragungen von „local“ nach „local“ ausführen.



Für Dateiübertragungen von „local“ nach „local“ verwenden Sie ein entsprechendes Script (*copy*) oder geben Sie eine der beiden Dateien als „remote“ und den lokalen Rechner als Partner an (*transferAdmission* angeben).

Format

```
<transferFile compress?="none|byteRep|zip" writeMode?=
"replace|new|extend" transparentMode?="true|false"
dataEncryption?="yes|no|onlyDataIntegrity">
  comment?
  context?
  ( (fromRemoteFile toLocalFile) |
    (fromLocalFile toRemoteFile) |
    (fromRemoteFile toLocalTmpFile) |
    (fromLocalTmpFile toRemoteFile) )
  remoteSuccessScript?
  remoteFailureScript?
</transferFile>
```

Attribute

Name	Wert	Bedeutung
compress?	<u>none</u> byteRep zip	Der Standardwert ist <i>none</i> . Die Datei wird nicht komprimiert. Bei Angabe von <i>byteRep</i> werden gleiche Zeichen in Folge komprimiert. Bei Angabe von <i>zip</i> wird zip-komprimiert.
writeMode?	<u>replace</u> new extend	Der Standardwert ist <i>replace</i> . Eine existierende Datei wird überschrieben. Existiert die Datei nicht, wird sie angelegt. Bei Angabe von <i>extend</i> werden die Daten an die angegebene Datei angehängt. Existiert die Datei nicht, wird sie angelegt. Bei Angabe von <i>new</i> wird eine neue Datei erzeugt. Existiert bereits eine Datei mit diesem Namen, wird die Aktivität mit dem Fehler <i>ft_exist</i> abgebrochen.
transparentMode?	true <u>false</u>	Der Standardwert ist <i>false</i> . Die Datei wird standardmäßig übertragen. Bei Angabe von <i>true</i> wird transparent übertragen (z.B. bei Übertragungen von einem BS2000-System zu einem anderen BS2000-System über Windows-/Unix-System).
dataEncryption?	yes <u>no</u> onlyDataIntegrity	Der Standardwert ist <i>no</i> . Die Benutzerdaten werden nicht verschlüsselt. Bei Angabe von <i>yes</i> werden die Benutzerdaten verschlüsselt (Einstellungen siehe openFT-Benutzerhandbuch). Bei Angabe von <i>onlyDataIntegrity</i> wird nur die Datenintegrität überprüft.

Beispiele

1. Dateiübertragung mit *remoteSuccessScript* / *remoteFailureScript*

```

<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <context>
    <script id="everything ok">
      <unixScript><![CDATA[echo everything ok
        >frg_eis_15/status.txt]]>
      </unixScript>
      <windowsScript>
        <![CDATA[cmd /c "echo everything ok"
          >frg_eis_15\status.txt]]>
      </windowsScript>
    </script>
    <script id="something failed">
      <unixScript><![CDATA[echo something failed>
        frg_eis_15/status.txt]]>
      </unixScript>
      <windowsScript>
        <![CDATA[cmd /c echo something failed
          >frg_eis_15\status.txt]]>
      </windowsScript>
    </script>
    <partner id="remote" name="UnixP_1" systemType="unix">
      <transferAdmission>
        <ftacAdmission ftacAdmission="FTACADM1"/>
      </transferAdmission>
    </partner>
  </context>
  <sequence>
    <context>
      <faulthandler>
        <default>
          <executeScript ref="something failed"/>
        </default>
      </faulthandler>
    </context>
    <transferFile>
      <fromRemoteFile name="bin.mp3">
        <partner ref="remote"/>
        <directory name="frg_eis_15"/>
      </fromRemoteFile>
      <toLocalFile name="bin.mp3">
        <directory name="frg_eis_15"/>
      </toLocalFile>
      <remoteSuccessScript ref="everything ok"/>
      <remoteFailureScript ref="something failed"/>
    </transferFile>
  </sequence>
</ftscript>

```

Beispiel 1 (Forts.)

```
<executeScript ref="everything ok"/>
</sequence>
<transferFile writeMode="extend">
  <fromRemoteFile name="status.txt" data="char">
    <partner ref="remote"/>
    <directory name="frg_eis_15"/>
  </fromRemoteFile>
  <toLocalFile name="status.txt">
    <directory name="frg_eis_15"/>
  </toLocalFile>
</transferFile>
</ftscript>
```

Es werden die Datei *bin.mp3* vom Partner *remote* in die lokale Datei *bin.mp3* übertragen. Anschließend wird die Datei *status.txt* mit folgendem Inhalt erzeugt:

- bei korrekter Übertragung

```
everything ok
everything ok
```

- im Fehlerfall

```
something failed
something failed
```

Wenn beim Zusammensetzen der Datei *status.txt* Fehler auftreten, führt dies zum Abbruch des Scripts.

2. Dateiübertragung von „remote“ nach „remote“

```
<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <context>
    <partner id="remote1" name="UnixP_1">
      <transferAdmission>
        <ftacAdmission ftacAdmission="FTACADM1"/>
      </transferAdmission>
    </partner>
    <partner id="remote2" name="WindowsP_1">
      <transferAdmission>
        <ftacAdmission ftacAdmission="FTACADM2"/>
      </transferAdmission>
    </partner>
  </context>
  <transferFile>
    <fromRemoteFile name="data.txt">
      <partner ref="remote1"/>
      <directory name="frg_eis_16"/>
      <autoDataSpec charPattern="*.txt" binPattern="*.dat *.mp3"/>
    </fromRemoteFile>
    <toLocalTmpFile id="tmp"/>
  </transferFile>
  <transferFile>
    <fromLocalTmpFile use="tmp"/>
    <toRemoteFile name="data.txt">
      <partner ref="remote2"/>
      <directory name="frg_eis_16"/>
    </toRemoteFile>
  </transferFile>
</ftscript>
```

Im Beispiel wird zuerst die Datei *data.txt* vom Partner *remote1* in eine temporäre Datei kopiert. Die temporäre Datei bekommt einen internen Namen. Die Endung der temporären Datei entspricht der Endung der zugehörigen *fromRemoteFile* Datei (hier **.txt*).

Die Konvertierung erfolgt mit *autoDataSpec charPattern* für Character-Zeichen, da die Endung der Datei dem Muster *.txt* entspricht (siehe [Abschnitt „autoDataSpec“ auf Seite 99](#)). Wenn das lokale System ein Windows-System ist, werden die Zeilenenden entsprechend konvertiert.

Bei der Übertragung der temporären Datei in das ferne System *remote2* werden die *autoDataSpec* Einstellungen übernommen, die beim Erzeugen der temporären Datei vorgenommen wurden. Diese werden vererbt, ebenso *maxRecSize* und Dateneigenschaften des Elements *fromRemoteFile*. Wenn das lokale System ein Windows-System ist, werden die Daten bei der zweiten Übertragung nicht konvertiert. Bei einem lokalen Unix-System wäre es umgekehrt. Bei der Übertragung in das lokale System werden keine Daten konvertiert, jedoch bei der zweiten Übertragung in das ferne System.

3. Verwendung von *faulthandler* und *executeScript* für lokale Scripts

Der *faulthandler* (siehe [Abschnitt „faulthandler“ auf Seite 69](#)) entspricht dem *localFailureScript*, das direkt auf *transferFile* folgende *executeScript* (siehe [Abschnitt „executeScript“ auf Seite 63](#)) entspricht dem *localSuccessScript*.

Schalten Sie einen *transferFile*-Auftrag mit *local*Script* wie im folgenden Beispiel dargestellt:

```
...
<sequence>
  <context>
    <faulthandler>
      <default>
        <executeScript>
          .... localFailureScript...
        </executeScript>
      </default>
    </faulthandler>
  </context>
  <transferFile>
    ....
  </transferFile>
  <executeScript>
    ... localSuccessScript...
  </executeScript>
</sequence>
...
```

Diese *sequence* kann auch in einer *parallel* oder *foreach - parallel* Anweisung stehen.

autoDataSpec

Mit *autoDataSpec* bestimmen Sie die Übertragungsart für die Datei, in der das Element angegeben wurde.

Ist der Datentyp einer Datei nicht bekannt, wird dieser anhand des Dateinamens über Muster (Pattern)-Erkennung bestimmt.

Als Pattern ist derzeit nur **.xxx* zulässig, da nur die Dateieindung geprüft wird.

Als Dateiname wird der verwendet, der für die aktuelle Übertragung angegeben ist.



Wenn ein Unix-spezifischer Dateiname und ein Windows-spezifischer Dateiname angegeben ist, wird je nach Partnertyp der entsprechende Dateiname ausgewertet.

Die Dateitypen sind im openFT-Benutzerhandbuch beschrieben.

Datentyp	Format	Beispiel
char	Textformat	*.xml
bin	Binärformat	*.doc im Windows-System
user	Benutzerformat	

autoDataSpec beschreibt, welches Pattern welchem Datentyp zugeordnet ist.

Wenn mehrere Pattern für verschiedene Dateiformate auf einen Dateinamen passen, wird der Dateityp in der Reihenfolge *bin*, *char*, *user* bestimmt.

Wenn für die Datei nach der Dereferenzierung ein expliziter Datentyp angegeben ist, wird *autoDataSpec* nicht ausgewertet.

Einschränkungen

Als Pattern ist nur **.xxx* sinnvoll, wobei *xxx* eine beliebige Länge haben darf. Die Gesamtlänge von 512 Zeichen für den ganzen Ausdruck darf jedoch nicht überschritten werden. Es wird nur die Endung der Datei überprüft.

Format

```
<autoDataSpec ref?="ID"
  charPattern?="patternList" binPattern?="patternList"
  userPattern?="patternList" default?="char|bin|user" >
  comment?
</autoDataSpec>
```

Attribute

Name	Wert	Bedeutung
ref?	string	Referenz auf ein <i>autoDataSpec</i> Kontextobjekt
binPattern?	patternlist ¹	Wildcard-Pattern. Wenn das Pattern auf den angegebenen Dateinamen passt, wird die Datei im <i>bin</i> -Format übertragen.
charPattern?	patternlist ¹	Wildcard-Pattern. Wenn das Pattern auf den angegebenen Dateinamen passt, wird die Datei im <i>char</i> -Format übertragen.
userPattern?	patternlist ¹	Wildcard-Pattern. Wenn das Pattern auf den angegebenen Dateinamen passt, wird die Datei im <i>user</i> -Format übertragen.
default?	<u>char</u> bin user	Der Standardwert ist <i>char</i> . Gibt den Datentyp an, der angenommen wird, falls kein Pattern gefunden wurde.

¹ patternlist ist die Liste der Patterns, welche durch Leerzeichen getrennt werden.

Beispiel

```

<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <context>
    <list id="FileList">
      <file name="bin.mp3"/>
      <file name="text.txt"/>
    </list>
    <list id="HostList">
      <partner name="UnixP_1">
        <transferAdmission>
          <ftacAdmission ftacAdmission="FTACADM1"/>
        </transferAdmission>
      </partner>
      <partner name="WindowsP_1">
        <transferAdmission>
          <ftacAdmission ftacAdmission="FTACADM2"/>
        </transferAdmission>
      </partner>
    </list>
  </context>
  <foreach listRef="HostList" selectType="partner"
    contextObject="host">
    <foreach listRef="FileList" selectType="file"
      contextObject="file">
      <transferFile>
        <fromLocalFile ref="file">
          <directory name="W:/openFT/ftscript/Test/data/small"/>
          <autoDataSpec charPattern="*.txt"
            userPattern="*.tab *.dat" default="bin"/>
        </fromLocalFile>
        <toRemoteFile ref="file">
          <partner ref="host"/>
          <directory name="frg_eis_17"/>
        </toRemoteFile>
      </transferFile>
    </foreach>
  </foreach>
</ftscript>

```

Im Beispiel wird die Dateiliste *FileList* der lokalen Dateien auf alle Rechner der *HostList* übertragen. Dabei werden gemäß der *autoDataSpec* Angabe die Dateiübertragungsmodi aktiviert. **.txt*-Dateien werden entsprechend dem Zielsystemtyp konvertiert. Alle unbekannt-Dateitypen werden binär übertragen. Für die *user*-Übertragung sind **.tab* und **.dat* definiert.

fromLocalFile

Mit *fromLocalFile* geben Sie die lokale Quelldatei für die Dateiübertragung an.

Die Übertragungsart (Datentyp) kann aus dem Dateinamen mit *autoDataSpec* abgeleitet werden (siehe [Abschnitt „autoDataSpec“ auf Seite 99](#)).

Dateinamen dürfen maximal 512 Zeichen lang sein. Dies wird vor der Ausführung des Ftscript überprüft. Die Länge des resultierenden Dateinamens (*directory* und *filenames*) ist durch das angesprochene Betriebssystem und die openFT-Version begrenzt. Die Länge ergibt sich aus der Anzahl der Zeichen plus 1 Zeichen, wenn *directory* nicht mit einem „/“ endet. Die Überprüfung findet während des Ftscript-Laufes statt.

Weitere Informationen dazu entnehmen Sie dem [Abschnitt „Dateinamen-Attribute“ auf Seite 20](#).

Einschränkungen

fromLocalFile darf nach der Dereferenzierung keine Partnerangabe besitzen.

Format

```
<fromLocalFile ref?="ID" data?="auto|char|bin|user"
    recordFormat?="std|undef|var|fix"
    maxRecSize?="int" ccsname?="string" filenames>
  comment?
  directory?
  autoDataSpec?
</fromLocalFile
```

Attribute

Name	Wert	Bedeutung
ref?	string	Referenz auf ein anderes Dateiojekt. Von diesem werden <i>partner</i> und <i>directory</i> übernommen, wenn Sie die Elemente hier nicht angegeben haben (siehe Abschnitt „Referenzierung“ auf Seite 15).
data?	<code>auto</code> <code>char</code> <code>bin</code> <code>user</code>	Der Standardwert ist <i>auto</i> . Gibt den Datentyp an. Bei Angabe von <i>char</i> , <i>bin</i> oder <i>user</i> wird die <i>autoDataSpec</i> Angabe ignoriert (siehe Abschnitt „autoDataSpec“ auf Seite 99).

Name	Wert	Bedeutung
recordFormat?	<code>std</code> <code>undef</code> <code>var</code> <code>fix</code>	Der Standardwert ist <i>std</i> . Gibt das Satzformat an. Es gilt die Standardzuordnung von openFT (<i>data=bin</i> -> <i>undef</i> , sonst <i>var</i>) Bei Angabe von <i>undef</i> ist das Satzformat undefiniert, z.B. für Binärformate. Bei Angabe von <i>var</i> ist das Satzformat variabel, z.B. für Textformate (ein Datensatz entspricht einer Zeile; Zeilen sind unterschiedlich lang). Bei Angabe von <i>fix</i> ist das Satzformat fest, z.B. <code>f80</code> (mit <i>recordFormat=fix</i> und <i>maxRecSize=80</i>).
maxRecSize?	1-65535 (openFT ≤ V11: 1-32756 bzw. 1-32767)	Gibt die Satzgröße an. Standardmäßig gilt der Wert von openFT (siehe openFT-Benutzerhandbuch). Bei <i>data=char</i> gibt <i>maxRecSize</i> die Länge einer Zeile an (bis CR/LF). Dateien mit einer Zeilenlänge größer als 65535 Bytes müssen mit <i>data=bin</i> übertragen werden. Ein Ftscrip, das beispielsweise einen, in openFT V11.0 unzulässigen <i>maxRecSize</i> -Wert verwendet, läuft mit openFT V11.0 nicht ab.
ccsname?	string max. 8 Zeichen	Angabe des Coded-Character-Set (siehe openFT-Benutzerhandbuch).
filenames	siehe Abschnitt „Dateinamen-Attribute“ auf Seite 20 .	

Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <transferFile>
    <fromLocalFile name="bin.mp3">
      <directory name="W:/openFT/ftscript/Test/data/small"/>
      <autoDataSpec charPattern="*.txt" userPattern="*.tab *.dat"
        default="bin"/>
    </fromLocalFile>
    <toRemoteFile name="bin.mp3">
      <partner name="UnixP_1">
        <transferAdmission>
          <ftacAdmission ftacAdmission="FTACADM1"/>
        </transferAdmission>
      </partner>
      <directory name="frg_eis_18"/>
    </toRemoteFile>
  </transferFile>
</ftscript>
```

Im Beispiel wird die lokale Datei *bin.mp3* in das ferne System *UnixP_1* unter dem Namen *bin.mp3* übertragen. Es ist keine Übertragungsart oder *autoDataSpec* angegeben. Die Übertragungsart ist *bin* (Standardwert von *autoDataSpec*).

fromLocalTmpFile

Mit *fromLocalTmpFile* geben Sie eine temporäre Quelldatei für die Dateiübertragung von „remote“ nach „remote“ an (siehe [„Arten der Dateiübertragung“ auf Seite 93](#)).

Die temporäre Datei ist nur eine Zwischenablage unter der lokalen Kennung. Diese temporäre Datei wird automatisch gelöscht.

Es gelten implizit die Angaben *data*, *maxRecSize*, *recordFormat*, *ccsname* und *autoDataSpec* der *fromRemoteFile toLocalTmpFile* Aktivität. Bei der Erzeugung eines *TmpFiles* aus *fromRemoteFile* werden diese Angaben vererbt.

Einschränkungen

fromLocalTmpFile können Sie nur nach einer *fromRemoteFile toLocalTmpFile* Aktivität verwenden, weil die Datei *TmpFile* bereits existieren und im aktuellen Kontext zugänglich sein muss. Siehe dazu [Abschnitt „fromRemoteFile“ auf Seite 107](#) und [Abschnitt „toLocalTmpFile“ auf Seite 113](#).

Format

```
<fromLocalTmpFile use="tmpID" />
```

Attribute

Name	Wert	Bedeutung
use	string	Verweis auf eine im aktuellen Kontext zugängliche Datei <i>TmpFile</i> . Diese Datei muss zuvor bei einem <i>transferFile</i> mit <i>toLocalTmpFile</i> erzeugt worden sein.

Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<ftscript version="1">
  <context>
    <partner id="FileServer" name="UnixP_1">
      <transferAdmission>
        <ftacAdmission ftacAdmission="FTACADM1"/>
      </transferAdmission>
    </partner>
    <list id="RemoteHostList">
      <partner name="UnixP_1">
        <transferAdmission>
          <ftacAdmission ftacAdmission="FTACADM1"/>
        </transferAdmission>
      </partner>
      <partner name="WindowsP_1">
        <transferAdmission>
          <ftacAdmission ftacAdmission="FTACADM2"/>
        </transferAdmission>
      </partner>
    </list>
  </context>
  <listDirectory listObject="FileList">
    <partner ref="FileServer"/>
    <baseDir name="frg_eis_19/datastore"/>
  </listDirectory>
  <foreach listRef="FileList" selectType="file"
    contextObject="File" execute="parallel">
    <transferFile>
      <fromRemoteFile ref="File">
        <partner ref="FileServer"/>
        <directory name="frg_eis_19/datastore"/>
      </fromRemoteFile>
      <toLocalTmpFile id="tmpFile"/>
    </transferFile>
    <foreach listRef="RemoteHostList" selectType="partner"
      contextObject="RemoteHost" execute="parallel">
      <transferFile>
        <fromLocalTmpFile use="tmpFile"/>
        <toRemoteFile ref="File">
          <partner ref="RemoteHost"/>
          <directory name="frg_eis_19/targetDir"/>
        </toRemoteFile>
      </transferFile>
    </foreach>
  </foreach>
</ftscript>
```

Im Beispiel werden die Dateien im Verzeichnis *frg_eis_19/datastore* auf die Rechner in das jeweilige Zielverzeichnis *frg_eis_19/targetDir* kopiert.

Mit *listDirectory* werden die Dateien des *FileServer* ermittelt.

Die erste *foreach* Aktivität durchläuft alle ermittelten Dateien des *FileServer*. Jede Datei wird in eine temporäre Datei *tmpFile* kopiert.

In der zweiten *foreach* Aktivität wird jede temporäre Datei in das Zielverzeichnis *frg_eis_19/targetDir* des *RemoteHost* kopiert.

Am Ende wird die zugehörige temporäre Datei *tmpFile* gelöscht.

fromRemoteFile

Mit *fromRemoteFile* geben Sie die ferne Quelldatei für die Dateiübertragung an.

Die Übertragungsart (Datentyp) kann aus dem Dateinamen mit *autoDataSpec* abgeleitet werden (siehe [Abschnitt „autoDataSpec“ auf Seite 99](#)).

Dateinamen dürfen maximal 512 Zeichen lang sein. Dies wird vor der Ausführung des Ftscript überprüft. Die Länge des resultierenden Dateinamens (*directory* und *filenames*) ist durch das angesprochene Betriebssystem und die openFT-Version begrenzt. Die Länge ergibt sich aus der Anzahl der Zeichen plus 1 Zeichen, wenn *directory* nicht mit einem „/“ endet. Die Überprüfung findet während des Ftscript-Laufes statt.

Weitere Informationen dazu entnehmen Sie dem [Abschnitt „Dateinamen-Attribute“ auf Seite 20](#).

Einschränkungen

fromRemoteFile muss nach der Dereferenzierung eine Partnerangabe besitzen.

Format

```
<fromRemoteFile ref?="ID" data?="auto|char|bin|user"
                recordFormat?="std|undef|var|fix"
                maxRecSize?="int" ccsname?="string" filenames>
  comment?
  partner?
  directory?
  autoDataSpec?
</fromRemoteFile
```

Attribute

Name	Wert	Bedeutung
ref?	string	Referenz auf ein anderes Dateiojekt. Von diesem werden <i>partner</i> und <i>directory</i> übernommen, wenn Sie die Elemente hier nicht angegeben haben (siehe Abschnitt „Referenzierung“ auf Seite 15).
data?	<code>auto</code> <code>char</code> <code>bin</code> <code>user</code>	Der Standardwert ist <i>auto</i> . Gibt den Datentyp an. Bei Angabe von <i>char</i> , <i>bin</i> oder <i>user</i> wird die <i>autoDataSpec</i> Angabe ignoriert (siehe Abschnitt „autoDataSpec“ auf Seite 99).
recordFormat?	<code>std</code> <code>undef</code> <code>var</code> <code>fix</code>	Der Standardwert ist <i>std</i> . Gibt das Satzformat an. Es gilt die Standardzuordnung von openFT (<i>data=bin</i> -> <i>undef</i> , sonst <i>var</i>) Bei Angabe von <i>undef</i> ist das Satzformat undefiniert, z.B. für Binärformate. Bei Angabe von <i>var</i> ist das Satzformat variabel, z.B. für Textformate (ein Datensatz entspricht einer Zeile; Zeilen sind unterschiedlich lang). Bei Angabe von <i>fix</i> ist das Satzformat fest, z.B. <i>f80</i> (mit <i>recordFormat=fix</i> und <i>maxRecSize=80</i>).
maxRecSize?	1-65535 (openFT ≤ V11: 1-32756 bzw. 1-32767)	Gibt die Satzgröße an. Standardmäßig gilt der Wert von openFT (siehe openFT-Benutzerhandbuch). Bei <i>data=char</i> , gibt <i>maxRecSize</i> die Länge einer Zeile an (bis CR/LF). Dateien mit einer Zeilenlänge größer als 65535 Bytes müssen mit <i>data=bin</i> übertragen werden. Ein Ftscript, das beispielsweise einen, in openFT V11.0 unzulässigen <i>maxRecSize</i> -Wert verwendet, läuft mit openFT V11.0 nicht ab.
ccsname?	string max. 8 Zeichen	Angabe des Coded-Character-Set (siehe openFT-Benutzerhandbuch).
filenames	siehe Abschnitt „Dateinamen-Attribute“ auf Seite 20 .	

Beispiel

Siehe Beispiel auf [Seite 95](#).

remoteFailureScript

Mit *remoteFailureScript* führen Sie bei Misserfolg der Datenübertragung ein Script auf dem fernen System aus (siehe [Abschnitt „script“ auf Seite 89](#)).

Es wird die Funktionalität *-rf* des openFT-Auftrags genutzt (siehe Kommando *ft* im openFT-Benutzerhandbuch).



Achten Sie darauf, dass sich das *remoteFailureScript* und ein entsprechender *fault-handler* im Fscript (siehe [Abschnitt „faulthandler“ auf Seite 69](#)) gegenseitig beeinflussen können, da die Ausführung praktisch gleichzeitig (lokal und entfernt) stattfinden kann.

Siehe auch [Abschnitt „executeScript“ auf Seite 63](#).

Zusätzliche Hinweise

Das Script wird auch ausgeführt, wenn eine laufende Dateiübertragung durch einen *fault* abgebrochen wurde (intern wird bei einem *fault ft_cancel* abgesetzt, siehe [Abschnitt „fault“ auf Seite 66](#) und openFT-Benutzerhandbuch).

Bei einem *fault* wird nur für die laufenden Übertragungen das *remoteFailureScript* ausgeführt.

Format

```
<remoteFailureScript ref?="ID">
  comment?
  script?
  bs2000Script?
  unixScript?
  windowsScript?
  zosScript?
</remoteFailureScript>
```

Attribute

Name	Wert	Bedeutung
ref?	string	Referenz auf ein Script-Kontextobjekt. Ein dort beschriebener Partner wird nicht ausgewertet. Es wird der Partner der zugehörigen <i>transferFile</i> Aktivität verwendet. Die Angabe von <i>repeatable</i> in referenzierten Script-Objekten wird ignoriert.

Beispiel

Siehe Beispiel auf [Seite 95](#).

remoteSuccessScript

Mit *remoteSuccessScript* führen Sie bei Erfolg der Datenübertragung ein Script auf dem fernen System aus (siehe [Abschnitt „script“ auf Seite 89](#)).

Es wird die Funktionalität *-rs* des openFT-Auftrags genutzt (siehe Kommando *ft* im openFT-Benutzerhandbuch). Siehe auch [Abschnitt „executeScript“ auf Seite 63](#).

Format

```
<remoteSuccessScript ref?="ID">
  comment?
  script?
  bs2000Script?
  unixScript?
  windowsScript?
  zosScript?
</remoteSuccessScript>
```

Attribute

Name	Wert	Bedeutung
ref?	string	Referenz auf ein Script-Kontextobjekt. Ein dort beschriebener Partner wird nicht ausgewertet. Es wird der Partner der zugehörigen <i>transferFile</i> Aktivität verwendet. Die Angabe von <i>repeatable</i> in referenzierten Script-Objekten wird ignoriert.

Beispiel

Siehe Beispiel auf [Seite 95](#).

toLocalFile

Mit *toLocalFile* geben Sie die lokale Zielfile für die Dateiübertragung an.

Siehe auch [Abschnitt „file“ auf Seite 73](#).

Dateinamen dürfen maximal 512 Zeichen lang sein. Dies wird vor der Ausführung des Ftscript überprüft. Die Länge des resultierenden Dateinamens (*directory* und *filenames*) ist durch das angesprochene Betriebssystem und die openFT-Version begrenzt. Die Länge ergibt sich aus der Anzahl der Zeichen plus 1 Zeichen, wenn *directory* nicht mit einem „/“ endet. Die Überprüfung findet während des Ftscript-Laufes statt.

Weitere Informationen dazu entnehmen Sie dem [Abschnitt „Dateinamen-Attribute“ auf Seite 20](#).

Einschränkungen

toLocalFile darf nach der Dereferenzierung keine Partnerangabe besitzen.

Format

```
<toLocalFile ref?="ID" ccsname?="string" filenames>
  comment?
  partner?
  directory?
</toLocalFile>
```

Attribute

Name	Wert	Bedeutung
ref?	string	Referenz auf ein anderes Dateiojekt.
ccsname?	string max. 8 Zeichen	Angabe des Coded-Character-Set (siehe openFT-Benutzerhandbuch).
filenames	siehe Abschnitt „Dateinamen-Attribute“ auf Seite 20 .	

Beispiel

Siehe Beispiel auf [Seite 95](#).

toLocalTmpFile

Mit *toLocalTmpFile* geben Sie eine temporäre Zielfeile für die Dateiübertragung von „remote“ nach „remote“ an (siehe [„Arten der Dateiübertragung“ auf Seite 93](#)).

Die temporäre Datei ist nur eine Zwischenablage unter der lokalen Kennung. Die Datei wird einem Dateiojekt mit der angegebenen Id zugeordnet. Dieses Dateiojekt wird in den Eltern-Kontext des aktuellen *transferFile* Elements eingeblenet und darf dort noch nicht existieren.

Die Datei bekommt einen internen Namen. Die Endung der Datei wird aus dem Dateinamen der zugehörigen *fromRemoteFile* Datei ermittelt. Hat diese keine Endung, hat auch die erzeugte temporäre Datei keine Endung. Diese Datei wird im Benutzerspeicher abgelegt. Wenn der Speicherplatz für den Benutzer begrenzt ist (Disk Quota), kann es bei der Erzeugung von temporären Dateien zu einem Überschreiten des Limits kommen. Die *transferFile* Aktivität wird mit dem Fehlercode *ft_err_LOCERR_MEM* abgebrochen.

Das Dateiojekt erbt die Angaben *data*, *maxRecSize*, *recordFormat*, *ccsname* und *autoDataSpec* der zugehörigen *fromRemoteFile* Aktivität. Diese werden dann beim Aufruf der *fromLocalTmpFile* Aktivität wieder verwendet (siehe [Abschnitt „fromLocalTmpFile“ auf Seite 104](#)).

Die temporäre Datei wird automatisch gelöscht, sobald der Kontext, in dem das Dateiojekt definiert wurde, verlassen wird.

Siehe auch [Abschnitt „file“ auf Seite 73](#).

Einschränkungen

1. Auf die erzeugte temporäre Datei können Sie nur mit *use="tmpID"* aus einem *fromLocalTmpFile* zugreifen.
2. Die *transferFile fromRemoteFile* Aktivität muss abgeschlossen sein, bevor mit *transferFile toRemoteFile* auf die temporäre Datei zugegriffen wird. Verwenden Sie diese Aktivitäten daher in einer sequenziellen Abfolge, oder betten Sie sie, gegebenenfalls in eine *<sequence>* ein.

Format

```
<toLocalTmpFile id="tmpID"/>
```

Attribute

Name	Wert	Bedeutung
id	string	Id der temporären Datei im Eltern-Kontext des zugehörigen <i>transferFile</i> Elements. Auf diese Id können Sie nur mit <i>use="tmpID"</i> aus einem <i>fromLocalTmpFile</i> zugreifen.

Beispiel

Siehe Beispiel im [Abschnitt „fromLocalTmpFile“](#) auf Seite 105.

toRemoteFile

Mit *toRemoteFile* geben Sie die ferne Zielfile für die Dateiübertragung an.

Siehe auch [Abschnitt „file“ auf Seite 73](#).

Dateinamen dürfen maximal 512 Zeichen lang sein. Dies wird vor der Ausführung des Ftscript überprüft. Die Länge des resultierenden Dateinamens (*directory* und *filenames*) ist durch das angesprochene Betriebssystem und die openFT-Version begrenzt. Die Länge ergibt sich aus der Anzahl der Zeichen plus 1 Zeichen, wenn *directory* nicht mit einem „/“ endet. Die Überprüfung findet während des Ftscript-Laufes statt.

Weitere Informationen dazu entnehmen Sie dem [Abschnitt „Dateinamen-Attribute“ auf Seite 20](#).

Einschränkungen

toRemoteFile muss nach der Dereferenzierung eine Partnerangabe besitzen.

Format

```
<toRemoteFile ref?="ID" ccsname?="string" filenames>
  comment?
  partner?
  directory?
</toRemoteFile>
```

Attribute

Name	Wert	Bedeutung
ref?	string	Referenz auf ein anderes Dateiojekt.
ccsname?	string max. 8 Zeichen	Angabe des Coded-Character-Set (siehe openFT-Benutzerhandbuch).
filenames	siehe Abschnitt „Dateinamen-Attribute“ auf Seite 20 .	

Beispiel

Siehe Beispiel auf [Seite 95](#).

5 Fehlermeldungen

Eine Beschreibung und Auflistung der „schweren“ Ftscript-Fehlercodes entnehmen Sie dem [Abschnitt „Schwere“ Ftscript-Fehlercodes](#) auf Seite 24.

Die hier aufgelisteten Fehlercodes können Sie in der *case*-Anweisung der *faulthandler* Aktivität verwenden. Den internen Code finden Sie gegebenenfalls im Ausgabeparameter *ActivityObject* der *ftshwact*-Ausgabe.

Der interne Code hat den Aufbau *ft_codexxxx*, wobei *xxxx* die openFT-Fehlernummer ist.

Der Ftscript-Fehlercode wird den openFT-Fehlernummern zugeordnet (siehe auch openFT-Benutzerhandbuch, Kommando *ft help*).

Interner Code	Fehlercode	Beschreibung
ft_activeDirNotExists	ft_resource	Script-Auftragsverzeichnis existiert nicht. Wiederanlauf nicht möglich.
ft_alarmException	ft_error	Fehler in Alarmbehandlung. Wird ignoriert.
ft_alarmFailed	ft_error	Slowpoll: Es ist ein unbehandelter Fehler aufgetreten.
ft_callFtRuntime	ft_error	Laufzeitfehler beim Aufruf von FT.
ft_callSecurity	ft_access	Zugriffsfehler beim Löschen einer Datei.
ft_cancelCmdError	ft_error	Abbruch wurde mit unbekanntem Fehler beendet.
ft_cancelError	ft_error	Fehler beim Abbruch.
ft_cancelUnexpectedState	ft_error	Auftrag beendet, jedoch falscher Status. Abbruch erwartet.
ft_cantCreateJobListener	ft_resource	Kommunikation mit openFT gestört.
ft_cantInitializeJob	ft_error	Auftrag kann nicht initialisiert werden.
ft_cantWriteLogData	ft_resource	Logdaten konnten nicht geschrieben werden.
ft_circleRef	ft_reference	Zirkuläre Referenz nicht erlaubt.
ft_closeOrderQueue	ft_resource	Order-Queue konnte nicht ordentlich geschlossen werden.

Interner Code	Fehlercode	Beschreibung
ft_code1038	ft_ignore	Auftrag <Auftrags-Id> ist in der Beendigungsphase und kann nicht mehr gelöscht werden.
ft_code108	ft_connection	Auftrag <Auftrags-Id>. Fernes System nicht verfügbar.
ft_code20	ft_notExist	<lokale Datei> nicht gefunden.
ft_code2014	ft_paramError	Keine Änderungen von Dateiattributen verlangt.
ft_code2015	ft_admin	openFT hat keine Berechtigung, für diesen Benutzer Aufträge zu bearbeiten.
ft_code2016	ft_notEmpty	Verzeichnis <lokale Datei> ist nicht leer.
ft_code2017	ft_localFileStructure	Dateiattribute passen nicht zu Auftragsparametern.
ft_code2018	ft_error	Attribute konnten nicht geändert werden.
ft_code2019	ft_cantCreate	<lokale Datei> konnte nicht erzeugt werden.
ft_code2021	ft_notExist	CCS Name unbekannt.
ft_code2022	ft_notExist	Übergeordnetes Verzeichnis nicht gefunden.
ft_code2023	ft_exist	<lokale Datei> existiert bereits.
ft_code2024	ft_notSupported	Übertragung von Datei-Generationsgruppen nicht unterstützt.
ft_code2025	ft_access	Fehler beim Zugriff auf <lokale Datei>.
ft_code2026	ft_paramError	Resultierender Dateiname <lokale Datei> zu lang.
ft_code2027	ft_paramError	Kein Datei- oder Verzeichnisname angegeben.
ft_code2028	ft_auth	Ungültiges Management-Kennwort.
ft_code2029	ft_access	<lokale Datei> nicht verfügbar.
ft_code2030	ft_notExist	Home-Verzeichnis nicht gefunden.
ft_code2031	ft_access	Umbenennen nicht möglich.
ft_code2032	ft_resource	Zu wenig Speicherplatz für <lokale Datei>.
ft_code2033	ft_notExist	Dateieigentümer unbekannt.
ft_code2034	ft_auth	Ungültiges Dateikennwort.
ft_code2036	ft_cantDelete	Schutzzeit der Datei noch nicht abgelaufen.
ft_code2037	ft_auth	<lokale Datei> ist schreibgeschützt.

Interner Code	Fehlercode	Beschreibung
ft_code2038	ft_localFile Structure	Dateistruktur nicht unterstützt.
ft_code2039	ft_syntax	Syntaxfehler im resultierenden Dateinamen <lokale Datei>.
ft_code2040	ft_notSupported	Transparente Dateiübertragung nicht unterstützt.
ft_code2042	ft_notSupported	Erweitern der Datei bei transparenter Übertragung nicht möglich.
ft_code2043	ft_auth	Zugriff auf <lokale Datei> ist unzulässig.
ft_code2044	ft_paramTooLong	Folgeverarbeitung zu lang.
ft_code2045	ft_auth	Berechtigung für Folgeverarbeitung ungültig.
ft_code2046	ft_auth	Lokale Zugangsberechtigung ungültig.
ft_code2047	ft_auth	Auftrag vom lokalen FTAC zurückgewiesen.
ft_code2048	ft_notSupported	Funktion für Protokoll <Partner Protokolltyp> nicht unterstützt.
ft_code2049	ft_notSupported	Ferne Folgeverarbeitung nicht unterstützt.
ft_code2070	ft_admin	Auftrag <Auftrags-Id>. openFT hat keine Berechtigung mehr, für diesen Benutzer Aufträge zu bearbeiten.
ft_code2071	ft_notSupported	Auftrag <Auftrags-Id>. Benutzerdaten-Verschlüsselung nicht installiert.
ft_code2072	ft_abort	Auftrag <Auftrags-Id> wurde gelöscht.
ft_code2073	ft_corrupt	Auftrag <Auftrags-Id>. Fehler bei der Verschlüsselung.
ft_code2074	ft_cantCreate	Auftrag <Auftrags-Id>. <lokale Datei> konnte nicht angelegt werden.
ft_code2075	ft_notExist	Auftrag <Auftrags-Id>. Übergeordnetes Verzeichnis nicht mehr gefunden.
ft_code2076	ft_access	Auftrag <Auftrags-Id>. Fehler bei Ein/Ausgabe auf <lokale Datei>.
ft_code2077	ft_access	Auftrag <Auftrags-Id>. Datei nun gegen Mehrfachzugriff gesperrt.

Interner Code	Fehlercode	Beschreibung
ft_code2078	ft_access	Auftrag <Auftrags-Id>. <lokale Datei> nicht mehr verfügbar.
ft_code2079	ft_notExist	Auftrag <Auftrags-Id>. <lokale Datei> nicht mehr gefunden.
ft_code2080	ft_notExist	Auftrag <Auftrags-Id>. Home-Verzeichnis nicht mehr gefunden
ft_code2081	ft_resource	Auftrag <Auftrags-Id>. <lokale Datei> bekommt keinen Speicherplatz mehr.
ft_code2082	ft_notExist	Auftrag <Auftrags-Id>. Dateieigentümer nicht mehr bekannt.
ft_code2083	ft_error	Auftrag <Auftrags-Id>. Fehler bei Vor-/Nachverarbeitung.
ft_code2084	ft_error	Auftrag <Auftrags-Id>. Exitcode <2> bei Vor-/Nachverarbeitung.
ft_code2085	ft_auth	Auftrag <Auftrags-Id>. Dateikennwort nicht mehr gültig.
ft_code2086	ft_auth	Auftrag <Auftrags-Id>. <lokale Datei> ist nun schreibgeschützt.
ft_code2087	ft_localFile Structure	Auftrag <Auftrags-Id>. Dateistrukturfehler.
ft_code2088	ft_error	Auftrag <Auftrags-Id>. NDMS-Fehler <2>.
ft_code2089	ft_recoveryFailed	Auftrag <Auftrags-Id>. Wiederanlauf fehlgeschlagen.
ft_code2090	ft_error	Auftrag <Auftrags-Id>. Fehler beim Beenden der Dateiübertragung.
ft_code2092	ft_auth	Auftrag <Auftrags-Id>. Zugriff auf <lokale Datei> ist nicht mehr zulässig.
ft_code2093	ft_error	Auftrag <Auftrags-Id>. FTAM-Fehler <2>.
ft_code2094	ft_cantDelete	Auftrag <Auftrags-Id>. Schutzzeit der Datei noch nicht abgelaufen.
ft_code2095	ft_notSupported	Auftrag <Auftrags-Id>. Erweitern der Datei bei transparenter Übertragung nicht möglich.
ft_code2096	ft_notSupported	Auftrag <Auftrags-Id>. Dateistruktur nicht unterstützt.

Interner Code	Fehlercode	Beschreibung
ft_code2109	ft_connection	Auftrag <Auftrags-Id>. Anfrage zum Verbindungsaufbau vom lokalen Transportsystem zurückgewiesen.
ft_code2110	ft_connection	Auftrag <Auftrags-Id>. Daten-Integritätsprüfung zeigt einen Fehler an.
ft_code2111	ft_connection	Verschlüsselung/Datenintegritätsprüfung nicht möglich. Verschlüsselung ausgeschaltet.
ft_code2112	ft_connection	Auftrag <Auftrags-Id>. Datenintegritätsprüfung wird vom Partner nicht unterstützt.
ft_code2113	ft_connection	Auftrag <Auftrags-Id>. Verschlüsselung der Benutzerdaten für diesen Auftrag nicht möglich.
ft_code2114	ft_connection	Auftrag <Auftrags-Id>. Identifikation des lokalen Systems vom fernen System '<Partner>' zurückgewiesen.
ft_code2115	ft_connection	Auftrag <Auftrags-Id>. Unterbrechung durch das ferne System.
ft_code2116	ft_connection	Lokale Anwendung '<1>' nicht definiert.
ft_code2117	ft_connection	Lokale Anwendung '<1>' nicht verfügbar.
ft_code2118	ft_connection	Auftrag <Auftrags-Id>. Authentifizierung des lokalen Systems fehlgeschlagen.
ft_code2119	ft_connection	Auftrag <Auftrags-Id>. Lokales System im fernen System unbekannt.
ft_code2120	ft_connection	Fernes System '<Partner>' unbekannt.
ft_code2121	ft_connection	Auftrag <Auftrags-Id>. Authentifizierung des Partners fehlgeschlagen.
ft_code2122	ft_connection	Auftrag <Auftrags-Id>. Verbindung abgelehnt oder abgebaut. Ursache <2>.
ft_code2123	ft_connection	Auftrag <Auftrags-Id>. Fehler <2> bei OSS-Aufruf.
ft_code2124	ft_connection	Auftrag <Auftrags-Id>. Keine freie Transportverbindung.

Interner Code	Fehlercode	Beschreibung
ft_code2152	ft_paramError	Auftrag <Auftrags-Id>. Fernes System: Kein Datei- oder Verzeichnisname angegeben.
ft_code2153	ft_auth	Auftrag <Auftrags-Id>. Fernes System: Ungültiges Management Kennwort.
ft_code2154	ft_access	Auftrag <Auftrags-Id>. Fernes System: Datei/Verzeichnis '<ferne Datei>' nicht verfügbar.
ft_code2155	ft_notExist	Auftrag <Auftrags-Id>. Fernes System: Datei/Verzeichnis '<ferne Datei>' nicht gefunden.
ft_code2156	ft_notExist	Auftrag <Auftrags-Id>. Fernes System: Home-Verzeichnis nicht gefunden.
ft_code2157	ft_access	Auftrag <Auftrags-Id>. Fernes System: Umbenennen nicht möglich.
ft_code2158	ft_resource	Auftrag <Auftrags-Id>. Fernes System: Zu wenig Speicherplatz für '<ferne Datei>'.
ft_code2159	ft_notExist	Auftrag <Auftrags-Id>. Fernes System: Dateieigentümer unbekannt.
ft_code2160	ft_auth	Auftrag <Auftrags-Id>. Fernes System: Ungültiges Dateikennwort.
ft_code2161	ft_cantDelete	Auftrag <Auftrags-Id>. Fernes System: Schutzzeit der Datei noch nicht abgelaufen.
ft_code2162	ft_auth	Auftrag <Auftrags-Id>. Fernes System: Datei/Verzeichnis '<ferne Datei>' ist schreibgeschützt.
ft_code2163	ft_remoteFile Structure	Auftrag <Auftrags-Id>. Fernes System: Dateistruktur nicht unterstützt.
ft_code2164	ft_syntax	Auftrag <Auftrags-Id>. Fernes System: Syntaxfehler im resultierenden Dateinamen.
ft_code2165	ft_notSupported	Auftrag <Auftrags-Id>. Fernes System: Transparente Dateiübertragung nicht unterstützt.
ft_code2166	ft_notSupported	Auftrag <Auftrags-Id>. Fernes System: Erweiterung der Datei bei transparenter Übertragung nicht möglich.

Interner Code	Fehlercode	Beschreibung
ft_code2167	ft_auth	Auftrag <Auftrags-Id>. Fernes System: Zugriff auf '<ferne Datei>' ist unzulässig.
ft_code2168	ft_paramTooLong	Auftrag <Auftrags-Id>. Fernes System: Folgeverarbeitung zu lang.
ft_code2169	ft_auth	Auftrag <Auftrags-Id>. Fernes System: Zugangsberechtigung ungültig.
ft_code2170	ft_notSupported	Auftrag <Auftrags-Id>. Fernes System: Funktion nicht unterstützt.
ft_code2195	ft_admin	Auftrag <Auftrags-Id>. Fernes System: openFT hat keine Berechtigung mehr für diesen Benutzer Aufträge zu bearbeiten.
ft_code2196	ft_abort	Auftrag <Auftrags-Id> wurde im fernen System gelöscht.
ft_code2197	ft_cantCreate	Auftrag <Auftrags-Id>. Fernes System: Datei/Verzeichnis '<ferne Datei>' konnte nicht angelegt werden.
ft_code2198	ft_notExist	Auftrag <Auftrags-Id>. Fernes System: Übergeordnetes Verzeichnis nicht mehr gefunden.
ft_code2199	ft_access	Auftrag <Auftrags-Id>. Fernes System: Fehler bei Ein/Ausgabe auf '<ferne Datei>'.
ft_code2200	ft_access	Auftrag <Auftrags-Id>. Fernes System: Datei nun gegen Mehrfachzugriff gesperrt .
ft_code2201	ft_access	Auftrag <Auftrags-Id>. Fernes System: Datei/Verzeichnis '<ferne Datei>' nicht mehr verfügbar.
ft_code2202	ft_notExist	Auftrag <Auftrags-Id>. Fernes System: Datei/Verzeichnis '<ferne Datei>' nicht mehr gefunden.
ft_code2203	ft_notExist	Auftrag <Auftrags-Id>. Fernes System: Home-Verzeichnis nicht mehr gefunden.
ft_code2204	ft_resource	Auftrag <Auftrags-Id>. Fernes System: Datei/Verzeichnis '<ferne Datei>' bekommt keinen Speicherplatz mehr.
ft_code2205	ft_notExist	Auftrag <Auftrags-Id>. Fernes System: Dateieigentümer nicht mehr bekannt.

Interner Code	Fehlercode	Beschreibung
ft_code2206	ft_error	Auftrag <Auftrags-Id>. Fernes System: Fehler bei Vor-/Nachverarbeitung.
ft_code2207	ft_error	Auftrag <Auftrags-Id>. Fernes System: Exitcode <2> bei Vor-/Nachverarbeitung.
ft_code2208	ft_auth	Auftrag <Auftrags-Id>. Fernes System: Dateikennwort nicht mehr gültig.
ft_code2209	ft_auth	Auftrag <Auftrags-Id>. Fernes System: Datei/Verzeichnis '<ferne Datei>' ist nun schreibgeschützt.
ft_code2210	ft_remoteFile Structure	Auftrag <Auftrags-Id>. Fernes System: Dateistrukturfehler.
ft_code2211	ft_error	Auftrag <Auftrags-Id>. Fernes System: NDMS-Fehler <2>.
ft_code2212	ft_recoveryFailed	Auftrag <Auftrags-Id>. Fernes System: Wiederanlauf fehlgeschlagen.
ft_code2213	ft_resource	Auftrag <Auftrags-Id>. Fernes System: Betriebsmittelengpass.
ft_code2214	ft_auth	Auftrag <Auftrags-Id>. Fernes System: Zugriff auf '<ferne Datei>' ist nicht mehr zulässig.
ft_code2215	ft_error	Auftrag <Auftrags-Id>. FTAM-Fehler <2>.
ft_code2216	ft_remoteFile Structure	Auftrag <Auftrags-Id>. Fernes System: Dateistruktur nicht unterstützt.
ft_code2217	ft_cantDelete	Auftrag <Auftrags-Id>. Fernes System: Schutzzeit der Datei noch nicht abgelaufen.
ft_code2218	ft_notSupported	Auftrag <Auftrags-Id>. Fernes System: Erweiterung der Datei bei transparenter Übertragung nicht möglich.
ft_code2226	ft_error	Inhalt der Monitordatei inkonsistent.
ft_code2227	ft_error	Monitordatei wird nicht von openFT verwendet.
ft_code2228	ft_error	Monitordatei nicht vorhanden.
ft_code236	ft_admin	Eingestellte Instanz '<1>' nicht mehr gefunden.
ft_code35	ft_access	Datei gegen Mehrfachzugriff gesperrt.

Interner Code	Fehlercode	Beschreibung
ft_code41	ft_resource	Kein Platz im Auftragsbuch frei.
ft_code700	ft_syntax	Parameter '<1>' und '<2>' dürfen nicht gleichzeitig angegeben werden.
ft_code701	ft_syntax	Eingabefehler.
ft_code702	ft_syntax	Parameterwert '<1>' zu lang.
ft_code703	ft_syntax	Pflichtparameter fehlt.
ft_code704	ft_syntax	Pflichtparameter '<1>' fehlt.
ft_code705	ft_syntax	Parameter '<1>' mehrfach angegeben.
ft_code706	ft_syntax	Parameter '<1>' kann nur zusammen mit '<2>' angegeben werden.
ft_code707	ft_syntax	Ungültiger Parameter '<1>'.
ft_code708	ft_syntax	Wertebereich des Parameters '<1>' nicht eingehalten.
ft_code709	ft_syntax	Zu viele Stellungsparameter.
ft_code710	ft_syntax	Fehlerhafter Parameterwert '<1>'.
ft_code750	ft_syntax	Kommando nicht bekannt.
ft_code751	ft_syntax	Kommandoname mehrdeutig bezüglich '<1>'.
ft_code752	ft_syntax	Abschließende Klammer fehlt bei Operand '<1>'.
ft_code753	ft_syntax	Fehlerhaftes Trennzeichen '<1>' nach Operand '<2>'.
ft_code755	ft_syntax	Listenwert des Operanden '<1>' mit Datentyp '<2>' nicht vereinbar.
ft_code756	ft_syntax	Struktureinleitender Operandenwert erforderlich für '<1>'.
ft_code757	ft_syntax	Wert des Operanden '<1>' mit Datentyp '<2>' nicht vereinbar.
ft_code758	ft_syntax	Schlüsselwortwert des Operanden '<1>' mehrdeutig bezüglich '<2>'.
ft_code759	ft_syntax	Zu viele schließende Klammern.
ft_code760	ft_syntax	Geforderter Operand '<1>' nicht vorhanden.
ft_code762	ft_syntax	Operandenname '<1>' mehrdeutig bezüglich '<2>'.
ft_code763	ft_syntax	Operand '<1>' nicht bekannt.
ft_code764	ft_syntax	Mehrfache Angabe des Operanden '<1>'.

Interner Code	Fehlercode	Beschreibung
ft_code765	ft_syntax	Zu viele Listen-Elemente für Operand '<1>'.
ft_code766	ft_syntax	Zu viele Stellungen-Operanden.
ft_code767	ft_syntax	Zu viele Stellungen-Operanden für '<1>'.
ft_code780	ft_syntax	Interner Fehler: Zu wenig Operanden-Speicher.
ft_code781	ft_syntax	Interner Fehler: Zu tiefe Strukturschachtelung.
ft_code790	ft_syntax	Verfügbare Kommandos: '<1>'.
ft_code791	ft_syntax	Verfügbare Listenwerte: '<1>'.
ft_code792	ft_syntax	Verfügbare Operanden: '<1>'.
ft_code793	ft_syntax	Verfügbare Werte: '<1>'.
ft_code800	ft_error	Auftrag <Auftrags-Id>. Interner Fehler. Monitordatei nicht zugreifbar.
ft_code801	ft_error	Auftrag <Auftrags-Id>. Interner Fehler.
ft_code802	ft_error	Auftrag <Auftrags-Id>. Warnung: Inhalt der Monitordatei '<2>' inkonsistent.
ft_code803	ft_error	Auftrag <Auftrags-Id>. Folgeverarbeitung kann nicht gestartet werden.
ft_code804	ft_error	Auftrag <Auftrags-Id>. Inkonsistente Auftragsdaten.
ft_code850	ft_error	Interner Fehler. Monitordatei nicht zugreifbar.
ft_code851	ft_error	Interner Fehler.
ft_code852	ft_error	Interner Fehler. Eingestellte Instanz '<1>' inkompatibel.
ft_code853	ft_error	Nachladefehler. Error-Code <1>.
ft_code854	ft_error	Schreiben von Loggingsätzen nicht mehr möglich. Prozess beendet.
ft_code855	ft_error	Kein Speicherplatz mehr für interne Dateien vorhanden.
ft_code856	ft_error	Fehler bei OPS-Ausgabe.
ft_code857	ft_error	Fehler in Schlüsseldatei '<1>'.

Interner Code	Fehlercode	Beschreibung
ft_code858	ft_error	Interner Fehler. Setzen / Rücksetzen der Dateisperren nicht möglich.
ft_code859	ft_error	Funktion wird wegen Teleservice Einschränkung nicht unterstützt.
ft_code860	ft_error	Teleservice Einschränkung für FTAC wegen FT.
ft_code861	ft_error	Teleservice Einschränkung für <1>.
ft_code862	ft_error	Protokollstack '<1>' nicht installiert.
ft_code999	ft_error	openFT panic <1>. Abnormale Beendigung.
ft_compressModelllegal	ft_error	Fehlerhafter Compress Modus. Schema fehlerhaft?
ft_compressModeOutOfRange	ft_error	Fehlerhafte Angabe im Compress Modus. ftAPI Version?
ft_contextCantCreate	ft_error	Kontext konnte nicht angelegt werden.
ft_contextCantCreate2	ft_error	Kontext konnte nicht erzeugt werden.
ft_createDirectoryFailed	ft_resource	Verzeichnis konnte nicht angelegt werden oder es ist kein Verzeichnis.
ft_createDirectorySecurity	ft_access	Verzeichnis durfte nicht angelegt werden.
ft_createNoFilename	ft_paramError	Beim Erzeugen eines Verzeichnisses kein Name angegeben.
ft_createParamError	ft_error	Parameterfehler beim Anlegen eines Verzeichnisses.
ft_deleteFailed	ft_error	Lokale Datei/lokales Verzeichnis konnte nicht gelöscht werden.
ft_deleteNoFilename	ft_paramError	Beim Löschen wurde kein Verzeichnis- oder Dateiname angegeben.
ft_deleteOrderQueue	ft_resource	Order-Queue konnte nicht gelöscht werden.
ft_deleteParamError	ft_error	Parameter Fehler beim Löschen.
ft_deleteParents	ft_error	..' Verzeichnis wird nicht gelöscht.
ft_deleteSelf	ft_error	.' Verzeichnis wird nicht gelöscht.
ft_delparError	ft_error	Fehler in den internen <i>delete</i> -Parametern.
ft_directionModeOutOfRange	ft_error	Fehlerhafte Angabe für Direction. ftAPI Version?
ft_directoryElementUnknown	ft_error	Schema und Ftsript passen nicht zusammen. Es gibt kein solches Directory Element.

Interner Code	Fehlercode	Beschreibung
ft_doubleContextObject	ft_error	Kontextobjekt mehrfach initialisiert.
ft_doubleLocalTmpFileRef	ft_reference	Id für lokale Temporärdatei in diesem Kontext bereits vergeben.
ft_emptyEventQueue	ft_error	Interner Fehler: Leere Queue.
ft_emptyParallelQueue	ft_error	Interner Fehler: Leere Queue.
ft_emptyTransferAdmission	ft_paramError	Die TransferAdmission enthält keine user- oder FTAC-Admission.
ft_encryptionIllegal	ft_error	Fehlerhafte Angabe für Encryption. Schema fehlerhaft?
ft_encryptionOutOfRange	ft_error	Fehlerhafte Angabe für Encryption. ftAPI Version?
ft_err_CONNERR_NOCONN	ft_connection	keine freie Transportverbindung.
ft_err_CONNERR_NOTAVAIL	ft_connection	Das ferne System ist nicht verfügbar.
ft_err_CONNERR_UNKNOWN	ft_configuration	Das ferne System ist unbekannt.
ft_err_INT_CRFILE	ft_error	Fehler beim Erzeugen einer Datei.
ft_err_INT_FORK_<errno>	ft_error	Fehler bei Systemaufruf fork. errno ist der Wert der errno-Variablen, der durch den fehlerhaften Systemaufruf gesetzt wird.
ft_err_INT_INIT	ft_error	Der Server kann nicht initialisiert werden.
ft_err_INT_INTERNAL_FN	ft_panic	Sonstiger interner Fehler: nicht unterstützte Funktion.
ft_err_INT_INTERNAL_VERS	ft_panic	Sonstiger interner Fehler: nicht unterstützte Version der Datenstruktur.
ft_err_INT_MEM	ft_resource	Fehler bei der Speicheranforderung.
ft_err_INT_OPEN_<errno>	ft_error	Fehler bei Systemaufruf open. errno ist der Wert der errno-Variablen, der durch den fehlerhaften Systemaufruf gesetzt wird.
ft_err_INT_OPENDIR_<errno>	ft_error	Fehler bei Systemaufruf opendir. errno ist der Wert der errno-Variablen, der durch den fehlerhaften Systemaufruf gesetzt wird.
ft_err_INT_PIPE_<errno>	ft_error	Fehler bei Systemaufruf pipe. errno ist der Wert der errno-Variablen, der durch den fehlerhaften Systemaufruf gesetzt wird.
ft_err_INT_READ_<errno>	ft_error	Fehler bei Systemaufruf read. errno ist der Wert der errno-Variablen, der durch den fehlerhaften Systemaufruf gesetzt wird.

Interner Code	Fehlercode	Beschreibung
ft_err_INT_RMFILE_ <errno>	ft_error	Fehler bei Systemaufruf rmfile. errno ist der Wert der errno-Variablen, der durch den fehlerhaften Systemaufruf gesetzt wird.
ft_err_INT_SIGNAL_signal	ft_error	Befehl wurde durch signal unterbrochen. signal bezeichnet das Signal, das die Unterbrechung verursachte.
ft_err_INT_STAT_ <errno>	ft_error	Fehler bei Systemaufruf stat. errno ist der Wert der errno-Variablen, der durch den fehlerhaften Systemaufruf gesetzt wird.
ft_err_INT_SYSTEM_ <errno>	ft_error	Fehler bei Systemaufruf system. errno ist der Wert der errno-Variablen, der durch den fehlerhaften Systemaufruf gesetzt wird.
ft_err_INT_WRITE_ <errno>	ft_error	Fehler bei Systemaufruf write: errno ist der Wert der errno-Variablen, der durch den fehlerhaften Systemaufruf gesetzt wird. Wenn nicht alle Bytes geschrieben werden konnten, hat errno den Wert -1.
ft_err_LOCERR_EXIST	ft_exist	Die lokale Datei existiert bereits.
ft_err_LOCERR_FTAC	ft_access	Der Auftrag wurde vom lokalen FTAC abgewiesen.
ft_err_LOCERR_FTC_ <exit-status>	ft_panic	<exit-status> bezeichnet die Meldungsnummer des ftc-Kommandos (siehe Meldung 94 und Meldung 95 in der openFT-Meldungstabelle).
ft_err_LOCERR_INCONS	ft_localFile Structure	Die lokale Datei ist inkonsistent.
ft_err_LOCERR_MEM	ft_resource	Die lokale Datei bekommt keinen Speicher.
ft_err_LOCERR_NOACCESS	ft_access	Auf die lokale Datei kann nicht zugegriffen werden.
ft_err_LOCERR_NOCREAT	ft_cantCreate	Die lokale Datei kann nicht angelegt werden.
ft_err_LOCERR_NOTEXIST	ft_notExist	Die lokale Datei kann nicht gefunden werden.
ft_err_PAR_DIRAC_ <errno>	ft_error	errno bezeichnet den Wert der errno-Variablen, der durch den stat()-Aufruf gesetzt wurde. Die errno-Variable hat den Wert 0, wenn für das Dateiverzeichnis kein Schreibrecht vergeben ist.
ft_err_PAR_FTMSG_ <code>	Fehlercode gemäß ft_code<code>	<Meldung entsprechend ft_code<code>>.

Interner Code	Fehlercode	Beschreibung
ft_err_PAR_INVSESS	ft_panic	Die Sitzungsnummer (Session) ist ungültig.
ft_err_PAR_LEN	ft_notSupported	Der Name des Arbeitsverzeichnisses (workdir) ist zu lang.
ft_err_PAR_LEN_ACCOUNT	ft_paramTooLong	Parameter zu lang: ftamext->account.
ft_err_PAR_LEN_CRPWD	ft_paramTooLong	Parameter zu lang: ftamext->crpasswd.
ft_err_PAR_LEN_FPWD	ft_paramTooLong	Parameter zu lang: mgmtpasswd oder filepasswd.
ft_err_PAR_LEN_LEGALQ	ft_paramTooLong	Parameter zu lang: ftamext -> legalq.
ft_err_PAR_LEN_LOCFN	ft_paramTooLong	Parameter zu lang: locfn.
ft_err_PAR_LEN_LOCPR	ft_paramTooLong	Parameter zu lang: Summe der Längen von locsuccproc und locfailproc.
ft_err_PAR_LEN_REMACC	ft_paramTooLong	Parameter zu lang: remaccount.
ft_err_PAR_LEN_REMADM	ft_paramTooLong	Parameter zu lang: remadmis.
ft_err_PAR_LEN_REMFN	ft_paramTooLong	Parameter zu lang: fn or remfn.
ft_err_PAR_LEN_REMPR	ft_paramTooLong	Parameter zu lang: Summe der Längen von remsuccproc und remfailproc.
ft_err_PAR_LEN_REMPWD	ft_paramTooLong	Parameter zu lang: rempasswd.
ft_err_PAR_LEN_REMSYS	ft_paramTooLong	Parameter zu lang: remsys.
ft_err_PAR_MAND	ft_panic	<ul style="list-style-type: none"> - Die Parameterliste par wurde nicht angegeben. - Der Name des Arbeitsverzeichnisses (workdir) wurde nicht angegeben. - Der Ausgabebereich stat wurde nicht angegeben. - Der Ausgabebereich info wurde nicht angegeben (nur bei ft_show()).
ft_err_PAR_MAND_LOCFN	ft_paramError	locfn wurde nicht angegeben.
ft_err_PAR_MAND_REMSYS	ft_error	Das ferne System wurde nicht angegeben.
ft_err_PAR_NODIR	ft_panic	Der angegebene Name (workdir) bezeichnet kein Dateiverzeichnis.
ft_err_PAR_NOTERM	ft_panic	Der Auftrag ist noch aktiv.
ft_err_PAR_OPEN	ft_panic	In einem Programm wurde einer Sitzung bereits dasselbe Dateiverzeichnis (workdir) zugeordnet.

Interner Code	Fehlercode	Beschreibung
ft_err_PAR_REMOTE_NOACCESS	ft_access	Keine Berechtigung zum Löschen im fernen System. Keine Berechtigung zum Lesen der Attribute im fernen System.
ft_err_PAR_REMOTE_NOTEMPTY	ft_notEmpty	Das Verzeichnis im fernen System ist nicht leer.
ft_err_PAR_REMOTE_NOTEXIST	ft_notExist	Datei/Verzeichnis existiert nicht im fernen System.
ft_err_PAR_TERM	ft_ignore	Der Auftrag ist schon beendet.
ft_err_PAR_VALUE	ft_panic	Unbekannter Parameter/Parameter sind inkompatibel. Der Name für das Arbeitsverzeichnis (workdir) ist ungültig.
ft_err_PAR_VALUE_ACCESS	ft_panic	ungültiger Parameter: ftamext ->accessmode.
ft_err_PAR_VALUE_AVAIL	ft_panic	ungültiger Parameter: ftamext ->available.
ft_err_PAR_VALUE_CANTIME	ft_panic	ungültiger Parameter: cantime.
ft_err_PAR_VALUE_COMPR	ft_panic	ungültiger Parameter: compress.
ft_err_PAR_VALUE_DIR	ft_panic	ungültiger Parameter: direction.
ft_err_PAR_VALUE_ENCRYPT	ft_resource	ungültiger Parameter: encryption.
ft_err_PAR_VALUE_FPWD	ft_panic	ungültiger Parameter: mgmtpasswd.
ft_err_PAR_VALUE_FTYPE	ft_panic	ungültiger Parameter: filetype.
ft_err_PAR_VALUE_LOCCSN	ft_paramError	ungültiger Parameter: locccsn.
ft_err_PAR_VALUE_MAXREC	ft_paramError	ungültiger Parameter: maxrecsize.
ft_err_PAR_VALUE_PRIO	ft_panic	ungültiger Parameter: priority.
ft_err_PAR_VALUE_REMACC	ft_paramError	ungültiger Parameter: remaccount.
ft_err_PAR_VALUE_REMADM	ft_auth	ungültiger Parameter: remadmis oder ungültiger Parameter: remadm. Die Benutzerkennung/Zugangsberechtigung im fernen System ist ungültig.
ft_err_PAR_VALUE_REMCCSN	ft_paramError	ungültiger Parameter: remccsn.
ft_err_PAR_VALUE_REMFN	ft_paramError	ungültiger Parameter: remfn. Die angegebene Datei existiert nicht/der Zugriff ist nicht erlaubt.
ft_err_PAR_VALUE_REMPWD	ft_paramError	ungültiger Parameter: rempasswd.
ft_err_PAR_VALUE_REMSYS	ft_paramError	ungültiger Parameter: rem. Das angegebene ferne System ist unbekannt.

Interner Code	Fehlercode	Beschreibung
ft_err_PAR_VALUE_RFORM	ft_paramError	ungültiger Parameter: record format.
ft_err_PAR_VALUE_RID	ft_panic	Die Request-Id (rid) ist ungültig.
ft_err_PAR_VALUE_STARTTIME	ft_panic	ungültiger Parameter: starttime.
ft_err_PAR_VALUE_SYNC	ft_panic	ungültiger Parameter: synchron.
ft_err_PAR_VALUE_TRANSP	ft_panic	ungültiger Parameter: transparent.
ft_err_PAR_VALUE_WMODE	ft_panic	ungültiger Parameter: writemode.
ft_err_PAR_VERS	ft_panic	Die Version der Datenstruktur (Parameterliste oder Ausgabebereich) ist ungültig.
ft_err_REMERR_EXIST	ft_exist	Die ferne Datei existiert bereits.
ft_err_REMERR_INCONS	ft_remoteFile Structure	Die ferne Datei ist inkonsistent.
ft_err_REMERR_MEM	ft_cantCreate	Die ferne Datei bekommt keinen Speicher.
ft_err_REMERR_NOACCESS	ft_access	Auf die ferne Datei kann nicht zugegriffen werden.
ft_err_REMERR_NOCREAT	ft_cantCreate	Die ferne Datei kann nicht angelegt werden.
ft_err_REMERR_NOTEXIST	ft_notExist	Die ferne Datei kann nicht gefunden werden.
ft_err_REMERR_REMADM	ft_auth	Die ferne Zugangsberechtigung ist ungültig.
ft_errorEncoding	ft_resource	Code Cp850 nicht unterstützt. Erweiterte Sprachunterstützung für Java installieren.
ft_errorOnCancel	ft_error	Fehler beim Abbruch.
ft_errorReadingCsv	ft_resource	Benutzer-Informationen konnten nicht gelesen werden.
ft_errorReadingExec	ft_resource	Standardfehlerausgabe der Script-Ausführung konnte nicht gelesen werden.
ft_errorReadingFtsadmErr	ft_resource	Standardfehlerausgabe des ftsadm-Kommandos konnte nicht gelesen werden.
ft_errorWaitingForExec	ft_resource	Fehler beim Warten auf das Ende der Script-Ausführung.
ft_errorWaitingForFtsadm	ft_resource	Fehler beim Warten auf das Ende des ftsadm-Kommandos.
ft_execFailed	ft_error	Script-Ausführung konnte nicht gestartet werden.
ft_exist	ft_exist	Verzeichnis existiert schon.
ft_exit_<exitcode>	ft_script	Mit executeScript ausgeführtes Script beendete sich mit Fehlercode.

Interner Code	Fehlercode	Beschreibung
ft_exit_255	ft_execute	Fehler beim Ausführen von executeScript.
ft_featuresNotSupported	ft_error	Parser unterstützt notwendiges Feature nicht.
ft_ftsadmFailed	ft_error	IOException beim Anstarten von ftsadm.
ft_housekeepingError	ft_error	Housekeeper wurde mit unbekanntem Fehler beendet. Wird wiederholt.
ft_illegalJobState	ft_error	Ungültiger Status bei Auftragsende gefunden.
ft_illegalMode	ft_error	Ungültiger Modus.
ft_infoFileCantWrite	ft_resource	Info Datei kann nicht geschrieben werden.
ft_InterpreterNotAlive	ft_resource	Interpreter konnte nicht gestartet werden. Ausführung später.
ft_invalidStartState	ft_error	Ungültiger Status beim Starten.
ft_jobCantCreate	ft_resource	Der Auftrag konnte nicht erzeugt werden.
ft_jobCloseError	ft_resource	Fehler beim Beenden des Auftrags.
ft_jobExists	ft_error	Diesen Auftrag gibt es schon.
ft_jobFailed	ft_error	Es ist ein unbekannter Fehler aufgetreten.
ft_listDirectoryError	ft_resource	Verzeichnis kann nicht gelesen werden.
ft_listDirectoryFileInfo	ft_resource	Fehler beim Schreiben der persistenten Daten.
ft_listDirectoryNoDirectory	ft_notExist	Kein Verzeichnis.
ft_listDirectoryOutOfRangeError	ft_error	Bereichsfehler beim ListDirectory.
ft_listDirectoryParamError	ft_error	Parameterfehler beim ListDirectory.
ft_listDirectorySecurity	ft_access	Zugriffsfehler beim ListDirectory eines lokalen Verzeichnisses.
ft_listDirWriterCantCreate	ft_resource	Kann persistente ListDirectory Daten nicht schreiben.
ft_listElementUnknown	ft_error	Schema und Ftscrip passen nicht zusammen. Es gibt kein solches Element in einer Liste!
ft_listenerCantClose	ft_resource	Listener konnte nicht beendet werden.
ft_lockNotReadable	ft_resource	Interpreterprozess nicht erreichbar.
ft_logfileCantCreate	ft_resource	Logfile konnte nicht angelegt werden. Kein Logfile, kein Replay.
ft_logfileCantRead	ft_resource	Logfile kann nicht gelesen werden. Replay nicht möglich.

Interner Code	Fehlercode	Beschreibung
ft_logfileError	ft_resource	Logfile konnte nicht angelegt werden. Kein Logfile, kein Replay.
ft_logfileInvalid	ft_resource	Ungültiges Logfile.
ft_logfileIoError	ft_resource	IO-Fehler für Logfile.
ft_logfileNoSuchFile	ft_resource	Logfile konnte nicht gefunden werden.
ft_logFileNotFound	ft_resource	Logfile konnte nicht gefunden werden.
ft_logfileNotReadable	ft_resource	Logfile kann nicht gelesen werden. Replay nicht möglich.
ft_logfileWrongVersion	ft_error	Falsche Version des Logfiles. Dieser Auftrag ist nicht wiederanlauffähig.
ft_loggerCantClose	ft_resource	Logger konnte nicht beendet werden.
ft_logWrapperCantCreate	ft_resource	Logfile-Wrapper konnte nicht angelegt werden.
ft_logWrapperCantFind	ft_resource	Logfile-Wrapper konnte nicht gefunden werden.
ft_logWrapperCantWrite	ft_resource	Logfile-Wrapper konnte nicht geschrieben werden.
ft_mainLockError	ft_resource	Mainlock nicht setzbar. Interpreter wird beendet.
ft_mainLockIoError	ft_resource	Mainlock nicht setzbar. Interpreter wird beendet.
ft_mainLockNotSet	ft_resource	Mainlock nicht gesetzt. Interpreter wird beendet.
ft_nameGroupElementUnknown	ft_error	Schema und Ftsript passen nicht zusammen. Es gibt kein solches Namens-Attribut.
ft_nasty_error	ft_error	Es ist ein unbehandelter Fehler aufgetreten.
ft_noInterpreter	ft_resource	ServerProzess nicht rechtzeitig vorhanden.
ft_noKeyInLogfile	ft_error	Logfile-Eintrag ungültig. "key" nicht vorhanden. Wiederanlauf nicht mehr möglich.
ft_noPartner	ft_paramError	Für die ferne Datei wurde kein Partner angegeben.
ft_noRef	ft_reference	Referenz kann nicht aufgelöst werden.
ft_noRefForContextObject	ft_reference	Referenz kann nicht aufgelöst werden.
ft_noRefInFtsript	ft_reference	Referenz kann nicht aufgelöst werden.
ft_noScript	ft_resource	Es wurde kein Script angegeben.

Interner Code	Fehlercode	Beschreibung
ft_noSession	ft_error	openFT-Session konnte nicht erzeugt werden.
ft_notExist	ft_notExist	Datei oder Verzeichnis existiert nicht.
ft_noTransferAdmission	ft_paramError	In der Partner-Angabe fehlt die TransferAdmission.
ft_notRepeatable	ft_error	Ftscript-Auftrag nicht wiederholbar.
ft_noUserFaultCode	ft_paramError	Fehlercode darf nicht mit 'ft' beginnen.
ft_noUserScript	ft_paramError	Für executeScript wurde kein Script angegeben.
ft_noUtf8Support	ft_resource	UTF 8 wird nicht unterstützt.
ft_openParamError	ft_error	Fehler beim Erzeugen der openFT-Session.
ft_openParamError2	ft_error	ft_open ging schief, Parameterfehler.
ft_OrderQueueNotFound	ft_resource	Order-Queue konnte nicht gefunden werden.
ft_parseByteArray	ft_error	Parse- oder IO-Fehler.
ft_parseError	ft_error	Parse- oder IO-Fehler.
ft_portFileCantClose	ft_resource	PortFile konnte nicht ordentlich geschlossen werden.
ft_portNotFound	ft_resource	Kein Port gefunden. Ausführung später.
ft_readStatus	ft_error	Parameterfehler beim Status ermitteln.
ft_recordSize	ft_error	Fehlerhafte Angabe für maxRecSize. Schema fehlerhaft?
ft_recoveryCreateDirectory	ft_recoveryFailed	Wiederanlauf fehlgeschlagen. Siehe Handbuch.
ft_recoveryFailed	ft_recoveryFailed	Wiederanlauf fehlgeschlagen. Siehe Handbuch.
ft_replayIllegalEntry	ft_error	Beim Replay weder Fehlercode noch Request-Id gefunden.
ft_reqlistOutOfRange	ft_error	Fehler beim Abfragen der Request-Liste von openFT.
ft_reqstatParamError	ft_error	Fehlerhafter Parameter für reqstat. Interner Fehler.
ft_requestIdFormat	ft_error	Fehler im Format der Request-Id.
ft_requestIdIllegalFormat	ft_error	Fehlerhaftes Format für die Request-Id.
ft_requestInvalid	ft_error	Ungültiger Request.
ft_resumeForbidden	ft_error	Resume nicht möglich.
ft_resumeUnloggedTransfer	ft_resource	mehr als ein offener Filetransfer.

Interner Code	Fehlercode	Beschreibung
ft_rformIllegal	ft_error	Fehlerhafte Angabe für RecordFormat. Schema fehlerhaft?
Ft_rformOutOfRange	ft_error	Fehlerhafte Angabe für RecordFormat. ftAPI Version fehlerhaft?
ft_schemaConflict	ft_error	Unbekannte Aktivität, Schema defekt?
ft_schemaNotFound	ft_resource	Schemadatei <i>ftscript.xsd</i> nicht installiert.
ft_scriptElementUnknown	ft_error	Schema und Ftscript passen nicht zusammen. Es gibt kein solches Script.
ft_scriptIdNotGiven	ft_error	Script-Id nicht angegeben.
ft_scriptInfoFileCantCreate	ft_resource	Script-Info nicht erzeugbar.
ft_scriptInfoFileCantWrite	ft_resource	Script-Info nicht beschreibbar.
ft_scriptInfoFileCantWrite2	ft_resource	Script-Info nicht erzeugbar.
ft_serverSignalCommand	ft_error	Fehler in Auftragsannahme des ServerProzesses.
ft_sessionNotFound	ft_resource	keine gültige openFT-Session bekommen.
ft_signalingFailed	ft_resource	Signalisierung an Port ging schief. Ausführung später.
ft_socketClose	ft_resource	Socket kann nicht geschlossen werden.
ft_stateUnknown	ft_resource	Unbekannter Status.
ft_statusCantRead	ft_resource	Status nicht lesbar. Auftrag nicht ausführbar.
ft_statusCantWrite	ft_resource	kann Status nicht schreiben.
ft_termSocketClose	ft_resource	Socket kann nicht geschlossen werden.
ft_timestampCantCreate	ft_resource	Timestamp konnte nicht angelegt werden.
ft_tmpFileDelete	ft_resource	TmpFile konnte nicht gelöscht werden.
ft_tmpFileNotFound	ft_resource	Erwartetes TmpFile wurde nicht gefunden.
ft_traceCantCreate	ft_resource	Trace-Datei konnte nicht erzeugt werden. Kein Trace (Trace abschalten!)
ft_traceFileCreateError	ft_resource	Trace-Datei konnte nicht erzeugt werden. (Trace abschalten)
ft_traceFileNotFound	ft_resource	Trace-Datei konnte nicht gefunden werden.
ft_traceSerialization	ft_error	Trace-Daten korrupt.
ft_traceSyncFailed	ft_resource	Trace-Daten konnten nicht auf Platte geschrieben werden. Trace evtl. nicht vollständig.

Interner Code	Fehlercode	Beschreibung
ft_traceWrapperCantCreate	ft_resource	Trace-Datei Wrapper konnte nicht erzeugt werden.
ft_traceWrapperCreateError	ft_resource	Trace-Datei Wrapper konnte nicht erzeugt werden.
ft_traceWrapperNotFound	ft_resource	Trace-Datei Wrapper konnte nicht gefunden werden.
ft_transferFileIllegalSyntax	ft_error	remote/remote oder lokal/lokal nicht erlaubt. Schema defekt?
ft_transferMandatoryParam	ft_error	Parameterfehler beim ft_transfer.
ft_transparentModelIllegal	ft_error	Fehlerhafte Angabe im Transparent-Mode. Schema fehlerhaft?
ft_transparentModeOutOfRange	ft_error	Fehlerhafte Angabe im Transparent-Mode. ftAPI Version korrekt?
ft_unexpectedLogEntry	ft_error	Logfile ungültig. Erwarteter Eintrag nicht gefunden.
ft_unknownError	ft_error	Unbekannter Fehlercode.
ft_unlockOrderQueue	ft_resource	Order-Queue konnte nicht freigegeben werden.
ft_unresolvedTmpFile	ft_reference	Lokales TmpFile nicht gefunden. Referenz kann nicht aufgelöst werden.
ft_unwrap	ft_error	Unwrapping fehlgeschlagen.
ft_userFault	ft_userFault	Fehler-Aktivität im Ftscript aufgerufen.
ft_userStorageCantCreate	ft_resource	Grundlegende Daten können nicht initialisiert werden. EXIT, KEIN ERRORLOG!
ft_validatingFeaturesNotSupported	ft_error	Interner Fehler: Parser unterstützt notwendiges Feature für Validierung nicht.
ft_validationError	ft_error	Validation-, Parse- oder IO-Fehler.
ft_workDirectoryPiNotFound	ft_error	PI für Aufrufsverzeichnis im Script nicht gefunden.
ft_workDirUnusable	ft_resource	WorkDir kann nicht erzeugt werden.
ft_writeModelIllegal	ft_error	Fehlerhafte Angabe im Write-Mode. Schema fehlerhaft?
ft_writeModeOutOfRange	ft_error	Fehlerhafte Angabe im Write-Mode. Schema fehlerhaft?
ft_writeOrderQueue	ft_resource	Schreiben in die Order-Queue fehlgeschlagen.
ft_wrongJavaVersion	ft_resource	Die aktuelle Java-Version ist zu alt.

Die folgenden Fehlercodes sind für den *faulthandler* zulässig (siehe [Abschnitt „faulthandler“ auf Seite 69](#)).

Fehlercodes	Beschreibung
ft_abort	Kommando abgebrochen.
ft_access	Zugriffsfehler.
ft_admin	Administrationsfehler.
ft_auth	Authentifizierungsfehler.
ft_cantCreate	Datei/Verzeichnis konnte nicht erzeugt werden.
ft_cantDelete	Datei/Verzeichnis konnte nicht gelöscht werden.
ft_configuration	Konfiguration fehlerhaft.
ft_connection	Verbindungsfehler.
ft_corrupt	Datei/Daten/Verzeichnis verboten.
ft_error	Allgemeiner Fehler.
ft_exist	Datei/Verzeichnis existiert schon.
ft_execute	Fehler beim Ausführen von executeScript.
ft_localFileStructure	Fehler in der lokalen Datei.
ft_notEmpty	Verzeichnis ist nicht leer.
ft_notExist	Datei/Verzeichnis existiert nicht.
ft_notSupported	Funktionalität wird nicht unterstützt.
ft_panic	Schwerer interner Fehler.
ft_paramError	Parameterfehler.
ft_paramTooLong	Parameter zu lang.
ft_recoveryFailed	Wiederanlauf fehlgeschlagen.
ft_reference	Es wird ein nicht vorhandenes Ftscript-Objekt referenziert.
ft_remoteFileStructure	Ferne Dateistruktur fehlerhaft.
ft_resource	Ressourcenfehler (z.B. kein Speicherplatz, kein Hauptspeicher)
ft_script	Mit executeScript ausgeführtes Script beendete sich mit Fehlercode.
ft_syntax	Syntaxfehler.

Fachwörter

Querverweise sind *kursiv* gekennzeichnet.

Aktivität

Ein *openFT-Script* besteht aus Aktivitäten. Dies können Arbeitsanweisungen an openFT sein (z.B. *transferFile*, *deleteFile*) oder Anweisungen, die den Ablauf steuern (z.B. *parallel*, *foreach*).

Auftrag

hier: *openFT-Auftrag*

Auftragsbuch

Datei, die *asynchrone Aufträge* und ihre Bearbeitungszustände enthält. Das Auftragsbuch enthält auch die mit dem Kommando *fmodo* gesetzten Parameter.

Dateiübertragungsauftrag

openFT-Auftrag

Eigentümer eines openFT-Auftrags

ist die Benutzerkennung im *lokalen System* bzw. *fernen System*, mit der dieser *openFT-Auftrag* durchgeführt wird. Eigentümer ist immer die Benutzerkennung, unter der der Auftrag abgesetzt wurde, nicht diejenige unter der der Auftrag durchgeführt wird.

Fernes System

siehe *Partnersystem*

File-Transfer

Dateiübertragung

FTAC (File Transfer Access Control)

Bestandteil von openFT, der einen erweiterten Zugangsschutz bei Dateiübertragung und Dateimanagement bietet.

FTAM-1

Dokumententyp für Textdateien

FTAM-3

Dokumententyp für Binärdateien

FTAM-Dateiattribute

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. 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-Partner

Partnersystem, mit dem über FTAM-Protokolle kommuniziert wird.

.ftsc

Dateiendung einer *openFT-Script*-Datei.

Ftscript

Mehrere logisch voneinander abhängige openFT-Aufträge können mit openFT-Script zu einem Auftrag (Ftscript) zusammengefasst werden.

Interpreter

Hier das Programm, das einen *openFT-Script-Auftrag* ausführt. Der Interpreter steuert den automatischen *Wiederanlauf*.

Job

Folge von Kommandos, Anweisungen und Daten.

Kennwort (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 Benutzerkennungs-Kennwort dient zur Authentisierung des Benutzers. Es dient dem Zugangsschutz. Das Datei-Kennwort dient zur Überprüfung der Zugriffsberechtigung beim Zugriff auf eine Datei (Jobvariable). Es dient dem Zugriffsschutz.

Kontextobjekt

In dem Kontext einer *Aktivität* sind Kontextobjekte beschrieben. Jedes Kontextobjekt hat eine Id, die innerhalb eines Kontextes eindeutig sein muss. Über diese Id wird das Kontextobjekt angesprochen (referenziert).

Lokales System

Das FT-System, an dem der Anwender arbeitet.

openFT-Auftrag

Auftrag an ein FT-System, eine Datei von einem Sendesystem zu einem Empfangssystem zu übertragen und gegebenenfalls Folgeverarbeitungsaufträge zu starten.

openFT-Partner

Partnersystem, mit dem über openFT-Protokolle kommuniziert wird.

openFT-Script

Name des Produkts und der Sprache, in der die *openFT-Aufträge* formuliert werden.

openFT-Script-Auftrag

Die Abarbeitung eines *Ftscripts*.

Partnersystem

hier: FT-System, das zusammen mit dem *lokalen System* openFT-Aufträge durchführt.

System, fernes

siehe *Partnersystem*

System, lokales

siehe *lokales System*

TCP/IP (Transmission Control Protocol/Internet Protocol)

Verbreitetes Protokoll zur Datenübertragung (entspricht etwa den Schichten 3 und 4 des ISO-Referenzmodells, d.h. Netzwerk- und Transportschicht); wurde ursprünglich für das ARPANET (Rechnernetz des US-Verteidigungsministeriums) entwickelt, inzwischen de-facto-Standard.

Trace

Diagnosefunktion, die den Ablauf des FT-Betriebs protokolliert.

Transmission Control Protocol/Internet Protocol

siehe *TCP/IP*

Transport Name Service (TNS)

Dienst zur Verwaltung transportsystemspezifischer Eigenschaften. Einträge für *Partnersysteme* enthalten die Informationen zum jeweils verwendeten Transportsystem.

Wiederanlauf

Automatische Fortsetzung eines *openFT-Auftrags* nach einer Unterbrechung.

Stichwörter

A

- Aktivitäten [13, 14](#)
 - Eltern- und Kind [14](#)
 - externe [14](#)
 - interne [14](#)
- Änderungen
 - gegenüber der Vorgängerversion [10](#)
- Anweisungen siehe openFT-Script-Anweisungen
- Attribute [19](#)
 - Dateinamen [20](#)
 - Verzeichnisnamen [21](#)
- Aufbau eines Ftscripts [13](#)
- Auftragsbuch [141](#)
- Ausführung eines Ftscript [26](#)
- Ausführung eines openFT-Scripts [26](#)
- autoDataSpec [99](#)

B

- baseDir [52](#)
- Bausteine eines Ftscripts [13](#)

C

- case [71](#)
- comment [53](#)
- context [54](#)
- createDirectory [55](#)

D

- Darstellungsmittel [10](#)
- Dateinamen angeben [20](#)
- Dateinamen-Attribute [20](#)
- default [71](#)
- deleteDirectory [57](#)
- deleteFile [59](#)

- Dereferenzierung [16](#)
 - Beispiel [17](#)
 - Regeln [16](#)
- Diagnose-Informationen [27](#)
- directory [61](#)

E

- empty [62](#)
- Ende eines openFT-Scripts [26](#)
- executeScript [63](#)

F

- fault [66](#)
- faulthandler [13, 15, 69](#)
 - case [71](#)
 - default [71](#)
 - zulässige Fehlercodes [139](#)
- Fehler eines openFT-Scripts [26](#)
- Fehlerbehandlung [13, 22](#)
- Fehlercodes [22, 117](#)
 - normale [22, 23](#)
 - schwere [22, 24](#)
 - zulässig für faulthandler [139](#)
- Fehlermeldungen [22, 117](#)
- file [73](#)
- foreach [74](#)
- fromLocalFile [102](#)
- fromLocalTmpFile [104](#)
- fromRemoteFile [107](#)
- ftacAdmission [87](#)
- FTAM-Dateiattribute [142](#)
- Ftscript
 - Aufbau [13](#)
 - Ausführung [26](#)
 - Bausteine [13](#)

ftscript 78

Ftscript-Fehlercodes 22

I

Initialisierung 26

K

Kennwort 142

kernel group 142

Kommando siehe openFT-Script-Kommando

Kontext 13, 15

L

list 79

listDirectory 80

N

normale Fehlercodes 22, 23

O

openFT-Script

 Ausführung 26

 Diagnose-Informationen 27

 Ende 26

 Fehler 26

 Initialisierung 26

openFT-Script-Anweisungen 51

 Attribute 19

 baseDir 52

 comment 53

 context 54

 createDirectory 55

 Dateinamen-Attribute 20

 deleteDirectory 57

 deleteFile 59

 directory 61

 empty 62

 executeScript 63

 fault 66

 faulthandler 69

 Fehlerbehandlung 22

 file 73

 foreach 74

 ftscript 78

list 79

listDirectory 80

parallel 83

partner 85

script 89

sequence 91

transferFile 93

Übersicht 51

 Verzeichnisnamen-Attribute 21

openFT-Script-Kommando 29

 Übersicht 29

P

parallel 83

partner 15, 85

 ftacAdmission 87

 processingAdmission 86

 transferAdmission 87

 userAdmission 87

Password 142

processingAdmission 86

R

Readme-Datei 11

Referenzierung 13, 15

remoteFailureScript 109

remoteSuccessScript 111

S

schwere Fehlercodes 22, 24

script 89

Script-Anweisungen siehe openFT-Script-Anweisungen

Script-Kommando siehe openFT-Script-Kommando

security group 142

sequence 91

storage group 142

T

TCP/IP 143

toLocalFile 112

toLocalTmpFile 113

toRemoteFile 115

transferAdmission [87](#)
transferFile [93](#)
 autoDataSpec [99](#)
 fromLocalFile [102](#)
 fromLocalTmpFile [104](#)
 fromRemoteFile [107](#)
 remoteFailureScript [109](#)
 remoteSuccessScript [111](#)
 toLocalFile [112](#)
 toLocalTmpFile [113](#)
 toRemoteFile [115](#)

U

userAdmission [87](#)

V

Verzeichnisnamen angeben [21](#)
Verzeichnisnamen-Attribute [21](#)

W

Wiederanlauf [25](#), [144](#)
Wurzel-Element [13](#), [14](#)

Z

z/OS [63](#)
Zielgruppe [7](#)

