SWIFT for corporates

FileAct Implementation Guide for SCORE

This document describes the rules users must follow when sending or receiving files using FileAct in SCORE (Standardised Corporate Environment).

November 2013

Legal Notices

Preface

Purpose of this document

This document describes the rules users must follow when sending or receiving files using FileAct in SCORE (Standardised CORporate Environment).

Compared to previous version (May 2010), this version introduces the eStatements solution.

Intended audience

This document is for users of FileAct in SCORE.

Related documentation

The following documents contain information related to this implementation guide:

- *SWIFTNet Messaging Operations Guide* – this details the full list of all FileAct operational rules and parameters

  The content of this implementation guide specifies how those rules and parameters must be implemented specifically in the context of SCORE.
- *SWIFTNet PKI Certificate Administration Guide* – contains more information about RBAC
- *SWIFTNet Service Description* - contains more information about RBAC

Document conventions

This document uses the following typographical conventions:

Bold	Names of files, parameters, API calls, user logon, and logon groups References to a directory or a menu GUI elements and command names
Italics	Important information and document names
Courier	User input, directory paths, parameter values, place holders, and system output examples

Table of Contents

1 Overview 5
1.1 FileAct Transaction Rules 5
1.2 File Transfer Modes 5
1.3 Role-Based Access Control (RBAC) 5
1.4 File Transfer Features 5
2 File Transfer Fields 6
2.1 General Rules 6
2.2 Service Name 6
2.3 Requestor and Responder DN 6
2.4 Request Type Field 7
2.4.1 Request Types for Payments and Cash Management Files 7
2.4.2 Request Types for Bank Account Management Files 8
2.4.3 Request Types for Trade Files 9
2.5 FileInfo Field 10
2.5.1 FileInfo field for Payments and Cash Management Files 10
2.5.2 FileInfo field for Bank Account Management Files 10
2.5.3 FileInfo field for Trade Files 11
2.6 HeaderInfo field 11
2.6.1 HeaderInfo field for Payments and Cash Management Files 11
2.6.2 HeaderInfo field for Bank Account Management Files 12
2.6.3 HeaderInfo field for Trade Files 14
2.7 Other Optional Fields 15
2.7.1 RequestRef field 15
2.7.2 File Description field 15
ANNEX A: Format description codes 16
ANNEX B: Country specifics 20
FRANCE 20
RequestTypes 20
Request for Files specification 23
File transfer fields 23
Content of the RequestForFile 24
Content of the RequestForFileException 24
ANNEX C: eStatements Supplementary Information 28

1 Overview

1.1 FileAct Transaction Rules

This document describes the rules users must apply when preparing a file transfer request (send or receive) using FileAct in SCORE (Standardised CORporate Environment). These rules include, for example:

- the **file transfer features** (for example, delivery notification, store-and-forward)
- the **file transfer fields** (for example, Request Type field, Total Number of Transaction field)

The purpose of these rules is threefold:

1. To streamline the implementation of multiple FileAct links (that is, remove the need to systematically agree with every party on the meaning and use of the various parameters)
2. To remove the need to rely on specific filenames or file contents to enable the automatic processing of an incoming file
3. For bulk payments on SWIFTNet 6.1 onwards only: to ensure that users provide the necessary, and correctly structured, information to cater for the appropriate billing mechanism.

1.2 File Transfer Modes

SCORE caters for both the following transfer modes:

- **Real-time** mode
- **Store-and-forward** mode

Participants must agree bilaterally on which mode to use.

1.3 Role-Based Access Control (RBAC)

FileAct uses RBAC, SWIFT's central access control mechanism, for the SCORE service. Two RBAC roles have been defined and can be assigned to individual operators to have access to SCORE . These are:

- access_to_corp_fa - for real-time mode
- access_to_corp_fast - for store-and-forward mode

The delegator (normally a Security Officer) within each participating institution uses the Role Management screen of the SWIFTNet Online Operations Manager to maintain (assign / update/ remove) the relevant roles for each of the institution’s operators (SWIFTNet DN) that has access to this service.

1.4 File Transfer Features

The following table defines the presence of the transfer features, depending on the file transfer mode and the direction of the FileAct traffic:

	Delivery Notification		**Non-Repudiation**	**End-to-End Signature**
	Real-Time	**Store-and-forward**		
Corporate-to-Bank traffic	Mandatory	Optional	Mandatory	Mandatory
Bank-to-Corporate traffic	Optional	Optional	Optional	Mandatory

Note The End-to-End Signature feature provides authentication and integrity control between the sender's and the receiver's SWIFTNet Links.

2 File Transfer Fields

2.1 General Rules

The following file transfer fields provide the necessary information on the contents and purpose of the file in such a way that:

- the sender of the file does not require that the receiver processes the file in a specific manner based on the filename
- the receiver of the file does not require that the sender specifies the filename in a specific way (that is, the sender is free to define the filename)
- SWIFT can capture the necessary information required to later support the billing process
- the specification of all file transfer fields (except Request Type) may be in **upper or lower case-**

See the *Naming and Addressing Guide* and *SWIFTNet Messaging Operations Guide* for more information on the naming conventions that are part of the SWIFT User Handbook Online.

2.2 Service Name

For using FileAct in SCORE, users must use the following service names:

Purpose	Mode	Service name
Production	Real-time	swift.corp.fa
	Store-and-forward	swift.corp.fast
Pilot (test & training) operations	Real-time	swift.corp.fa!p
	Store-and-forward	swift.corp.fast!p
Testing (integration test bed - for application vendors only)	Real-time	swift.corp.fa!x
	Store-and-forward	swift.corp.fast!x

2.3 Requestor and Responder DN

These fields must be structured in one of the two following ways:

Type	Description
BIC only	Includes only the SWIFT registered BIC8 of the user that initiated the message exchange. For example: `o=corpbebb, o=swift` `o=bankbebb, o=swift`
BIC plus one extension	Includes the SWIFT registered BIC8 of the user that initiated the message, and one extension to identify the entity within the user that initiated the message exchange. For example: `u=treasury, o=corpbebb, o=swift` `ou=corpaccess, o=bankbebb, o=swift`

Extensions must only be used when they are meaningful to the recipient of the message. They designate a location, for example, a branch, a department, a business unit, a branded product, a person, a system, or an application.

Note Do not use extensions to describe the content of the message to be transferred. Use the Request Type field for this purpose.

The requestor DN used in this field must comply with the rules described in the *SWIFTNet Naming and Addressing Guide*.

2.4 Request Type Field

The **Request Type** field is mandatory and must be formatted as follows:

<area>.<syntax_and_format>.<description>

- **<area>** is a mandatory 4 letter code specifying the business area to which the file content belongs.
- **<syntax_and_format>** is also mandatory and identifies the syntax and specific format used.
- **<description>** is optional and provides additional information on the content of the file.

Depending on the business area, the content of the Request Type field differs:

2.4.1 Request Types for Payments and Cash Management Files

<area> is a code from the following list:

<area>	Content of file
pain	Credit transfers, and information about their status
	Direct debits, and information about their status
	Requests for cancellations
	Confirmations of requests for cancellations
camt	Credit/debit advises
	Intra-day statements
	End-of-period statements (day / week / month, etc)

<syntax_and_format> must be formatted as follows:

- `<nnn>.<mmm>.<vv>` when the file contains existing XML messages to the current SWIFTStandards, where `<area>.<nnn>.<mmm>.<vv>` refers to the message identifier, as defined by SWIFTStandards
- `fin.mt<nnn>` when the file contains FIN messages of the same type, and where `<nnn>` is the message type
- `fin.mt<n>xx` when the file contains multiple types of FIN messages, but all belong to the same FIN message category
- `fin.mtxxx` when the file contains multiple FIN messages involving more than one FIN message category
- `xxx.<format_description>` when the file contains a non-FIN and non-ISO 20022 format, and where the value of `<format_description>` must be according to the table in Annex A.

<description> may be defined bilaterally. SWIFT provides the following guidelines for the value of this field.

File content	<description>
Domestic credit transfer	dct
Cross-border credit transfer	xct
SEPA credit transfer	sct
SEPA direct debit	sdd
Domestic direct debits	ddd

File content	<description>
Statements	stm
Report	rep
eStatements (end-of-period)	eop
Information on ACH files	
File Acknowledgement	fak
Summary of submitted transactions	sts
Summary of returned items	sri
Detailed listing of returned items	dri
Detailed listing of confirmed items	dci

Examples:

Request type	File content
pain.001.001.02.sct	Sepa credit transfer initiation message (version 2) using SWIFTStandards MX
pain.fin.mt101	Bulked FIN MT101 messages
camt.xxx.cfonb120.stm	Statements formatted according to the CFONB format, fixed length 120 characters
camt.xxx.estmt.eop	eStatements for a designated reporting period

2.4.2 Request Types for Bank Account Management Files

<area> contains the code “acmt”, which covers:

<area>	Content of file
acmt	Bank account opening
	Bank account maintenance
	Bank account closing
	Reporting of bank account features

<syntax_and_format> must be formatted as follows:

- `<nnn>.<mmm>.<vv>` when the file only contains existing XML messages to the current SWIFTStandards, where `<area>.<nnn>.<mmm>.<vv>` refers to the message identifier, as defined by SWIFTStandards
- `xxx.<format_description>` when the file contains an XML message and attachments. In this case, `<format description>` contains the same part of the XML message identification: `<nnn>.<mmm>.<vv>.` The “`xxx`“ indicates the file contains additional non-SWIFT format

<description> is mandatory, if attachments are present, and contains the code “att”. The description of the attachments will be given in the header info field.

File content	<description>
Attachments	att

Examples:

Request type	Content of file
acmt.007.001.01	Xml message – account opening request - only

acmt.xxx.007.001.01.att	Xml message – account opening request – with attachments

2.4.3 Request Types for Trade Files

<area> is a code from the following list:

<area>	Content of file
tsrv	Letter of Credit application
	Request for Letter of Credit amendment
	Advice of Letter of Credit
	Guarantee application
	Request for Guarantee amendment
	Advice of Guarantee
	Standby application
	Request for Standby amendment
	Advice of Standby
tsmt	Invoices
	Bills of Lading
	Phyto-sanitary certificate
	Certificate of Origin

<syntax_and_format> must be formatted as follows:

- `fin.mt7xx` when the file contains multiple types of FIN Category 7 messages,
- `xxx` when the file contains a non-FIN format,

<description> is mandatory and contains one of the following values:

File content	<description>
Items pertaining to Letters of Credit	lettersofcredit
Items pertaining to Guarantees or Standbys	gteesstandbys
Items such as invoices and Bills of lading	tradedocuments

Examples:

Request type	Content of file
tsrv.fin.mt7xx.lettersofcredit	items pertaining to Letters of Credit using FIN Cat 7 format
tsrv.fin.mt7xx.gteesstandbys	items pertaining to Guarantees or Standbys using FIN Cat 7 format
tsrv.xxx.lettersofcredit	items pertaining to Letters of Credit using a non-FIN format
tsrv.xxx.gteesstandbys	items pertaining to Guarantees or Standbys using a non-FIN format
tsmt.xxx.tradedocuments	trade documents

Note When **<area>** is `tsmt`, then **<syntax_and_format>** always equals `xxx`, as there are no FIN standards for trade documents.

2.5 FileInfo Field

The **FileInfo** field provides additional structured information about the contents of the file, complementing the information provided in the **Request Type** field. The information contained in this field is of a technical nature (for example: compression, character set).

The different information elements must be separated by a semicolon. For example:

SwCompression=None;Test=Y

2.5.1 FileInfo field for Payments and Cash Management Files

The **FileInfo** field is **mandatory**.

For payments files, if the sender wants to benefit from bulk payment pricing, it must contain the following element as the first entry in the FileInfo field:

SwCompression=None

For eStatement files, FileInfo must contain the following element:

SwCompression=Zip

In some cases, users may bilaterally agree to sign the file at application level using a proprietary scheme (that is, not related to SWIFTNet). Then, this is indicated in the FileAct header as follows:

- **FileInfo** field will contain the following element:
  DataSign=1. Default value = 0 (that is, file is not signed on application level)
- **FileInfo** field can contain the following element:
  SignType=<value>, where value = name of signature method (for example, cms, autack)

Values specified in the FileInfo field must be case insensitive, meaning that applications reading this FileInfo field, must handle upper and lower case characters the same way. For example, all the following forms can be accepted: SWCOMPRESSION=ZIP, swcompression=zip and SwCompression=Zip, or any other combination of lowercase/uppercase.

To exchange additional information on the file, users may specify additional elements agreed upon bilaterally with correspondents (for example, FileType, ContractId). To maximise inter-operability, SWIFT recommends to only using such fields when necessary.

2.5.2 FileInfo field for Bank Account Management Files

The **FileInfo** field is **mandatory** and must contain the following element:

SwCompression=<value>, where **value** may be **None** or a designated keyword (see examples below)

In the context of bank account management, the need exists to concatenate multiple items when being transported into one single file. This will be the case when one XML message and its attachments are sent together. The FileInfo field will indicate whether and which compression algorithm has been used. This will enable the receiver of the file to extract the content of the file using the right application before actually processing the content of the file.

Examples:

1) No compression used by the sender (single file)

   SwCompression=None

2) Windows WINZip compression used by the sender

   SwCompression=Zip

3) Unix Gzip compression used by the sender

   SwCompression=Gzip

2.5.3 FileInfo field for Trade Files

The **FileInfo** field is **mandatory** and must contain the following element:

SwCompression=<value>, where **value** may be **None** or a designated keyword (see examples below)

In the context of trade, there is a requirement that the FileAct payload consist of only a single file in order to simplify cross referencing and linking of this file to a transaction that has been sent separately (e.g. via FIN). When multiple files need to be sent together, these files should be grouped into a single file using a compression tool such as WINZip. The FileInfo field is used to indicate whether the file sent over FileAct has been compressed and to specify the compression algorithm used. This will enable the receiver of a compressed file to extract the content using an appropriate application and to make it available in its source format for subsequent processing.

Examples:

1) No compression used by the sender (single file)

   SwCompression=None

2) Windows WINZip compression used by the sender

   SwCompression=Zip

3) Unix Gzip compression used by the sender

   SwCompression=Gzip

2.6 HeaderInfo field

The **HeaderInfo** field, available since SWIFTNet 6.1 (December 2007), can be used to specify processing instructions when sending information through FileAct.

See also the *SWIFT Standards MX General Information* for more information on using and implementing different service profiles in the **HeaderInfo** field, under the chapter titled: Structure of a SWIFTNet FileAct Message. This document is available on swift.com on www.swift.com > Ordering & Support > Documentation.

2.6.1 HeaderInfo field for Payments and Cash Management Files

The **HeaderInfo** field is not used for eStatements.

From 1 January 2008, SWIFT has been using a pricing scheme for Corporate-to-Bank payments sent over FileAct that is based on the total number of payment instructions contained in the file, and not the number of Kchars.

To benefit from this bulk payments pricing, file compression is **not** allowed (SwCompression = None) and the sender must indicate the number of payment instructions in the **HeaderInfo** field using an XML structure. The data element used to specify this number is detailed in the following table.

Element Name	XML Tag	Data Type	Description	Comment
Total Number Of Transactions	<TtlNbOfTxs>	Max15 Numeric Text	Total number of individual transactions contained in the file.	For Payments (pain and pacs) files, this field is used centrally at SWIFT to apply adequate pricing.

The use of the **HeaderInfo** field with the TotalNumberOfTransactions element is mandatory in order to benefit from the bulk payments pricing. In absence of this field, generic FileAct pricing terms apply.

XML Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns="urn:swift:xsd:ApplSpcfc.TxsCntr.01"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:swift:xsd:ApplSpcfc.TxsCntr.01"
elementFormDefault="qualified">
<xs:element name="ApplSpcfc" type="ApplSpcfc" />
 <xs:complexType name="ApplSpcfc">
  <xs:sequence>
   <xs:element name="TxsCntr" type="FileTransferHeaderTransactionsCounter1" />
  </xs:sequence>
 </xs:complexType>
 <xs:complexType name="FileTransferHeaderTransactionsCounter1">
  <xs:sequence>
   <xs:element name="TtlNbOfTxs" type="Max15NumericText" />
  </xs:sequence>
 </xs:complexType>
 <xs:simpleType name="Max15NumericText">
  <xs:restriction base="xs:string">
   <xs:pattern value="[0-9]{1,15}" />
  </xs:restriction>
 </xs:simpleType>
</xs:schema>
```

Example

In case a corporate sends a payment file containing 650 payment instructions, the corresponding **HeaderInfo** field will look as follows:

```
<ApplSpfc xmlns="urn:swift:xsd:ApplSpfc.TxsCntr.01" >
   <TxsCntr>
      <TtlNbOfTxs>650</TtlNbOfTxs>
   </TxsCntr>
</ApplSpfc>
```

See the *SWIFT Price List* for more information on pricing schemes for bulk payments from corporate entities to banks.

2.6.2 HeaderInfo field for Bank Account Management Files

The **HeaderInfo** field may contain, for information purposes, the name of the attachments – as per naming convention (still to be) agreed with the community – as well as their format using an XML structure. The data elements used to specify the name and type of the attachments are detailed in the below table. Standard FileAct pricing is used for sending EBAM related data over FileAct. Therefore, users do NOT need to use the **HeaderInfo** field for billing purposes.

Element Name	XML Tag	Data Type	Description	Comment
TypeOfFile	<Tp>	Max35Text	Identification of the format of an attached file to an EBAM XML message.	For EBAM (acmt) files, this field is used to identify to the receiver of the file the file-format of an attachment.
NameOfFile	<Nm>	Max70Text	Name of an attached file to an EBAM XML	For EBAM (acmt) files, this field is used to identify to the receiver of the

			message.	file the filename of an attachment.

XML Schema

```
<?xml version = „1.0“ encoding = „UTF-8“?>
<xs:schema xmlns=“urn:swift:xsd:ApplSpcfc.BAM.01“
targetNamespace=“urn:swift:xsd:ApplSpcfc.BAM.01“
xmlns:xs=http://www.w3.org/2001/XMLSchema elementFormDefault=“qualified“>
<xs:element name = „ApplSpcfc“ type = „ApplSpcfc“/>
 <xs:complexType name = „ApplSpcfc“>
  <xs:sequence>
   <xs:element name = „FileTrfHdrBAM“ type = „FileTransferHeaderBAMV01“/>
  </xs :sequence>
 </xs :complexType>
 <xs :complexType name = « File1 »>
  <xs :sequence>
   <xs :element name = « Tp » type = « Max35Text »/>
   <xs:element name = “Nm” type = “Max70Text”/>
  </xs:sequence>
 </xs:complexType>
 <xs:complexType name = “FileTransferHeaderBAMV01”>
  <xs:sequence>
   <xs:element name = “File” type = “File1” minOccurs = “0” maxOccurs =
“unbounded”/>
  </xs:sequence>
 </xs:complexType>
 <xs:simpleType name = “Max35Text”>
  <xs:restriction base = “xs:string”>
   <xs:maxLength value = “35”/>
   <xs:minLength value = “1”/>
  </xs:restriction>
 </xs:simpleType>
 <xs:simpleType name = “Max70Text”>
  <xs:restriction base = “xs:string”>
   <xs:maxLength value = “70”/>
   <xs:minLength value = “1”/>
  </xs:restriction>
 </xs:simpleType>
</xs:schema>
```

Example

In case two attachments – terms and conditions in pdf format and a copy of a passport in jpeg format – are attached, the corresponding **HeaderInfo** field will look as follows:.

```
<ApplSpcfc xmlns = “urn:swift:xsd:ApplSpcfc.BAM.01” xmlns:xsi =
“http://www.w3.org/2001/XMLSchema-instance”
   <FileTrfHdrBAM>
      <File>
         <Tp>terms</Tp>
         <Nm> BankTerms&Conditions.pdf </Nm>
      </File>
      <File>
         <Tp>passport</Tp>
         <Nm> Passport.jpg </Nm>
      </File>
```

```
    </FileTrfHdrBAM>
  </ApplSpcfc>
```

2.6.3 HeaderInfo field for Trade Files

The **HeaderInfo** field may be used to specify the name of the FileAct file and its related transaction reference number. Standard FileAct pricing is used for sending Trade data over FileAct. Therefore, users do NOT need to use the **HeaderInfo** field for billing purposes.

When the related transaction is a FIN message (e.g. MT 700), then this reference number represents the Transaction Reference Number (field 20) sent in the associated FIN message to which the FileAct file is linked. For the MT 798 trade envelope flows, this reference number is the first field in the MT798 index message.

When the related transaction is an MX message (e.g. tsmt.014.001.02), then this reference number represents the Transaction Identification (tag TxId) sent in the associated MX message to which the FileAct file is linked.

The **HeaderInfo** field is formatted as an XML structure, the data elements are detailed in the following table:

Element Name	XML Tag	Data Type	Description	Comment
NameOfFile	<Nm>	Max70Text	Name of the FileAct file.	For trade (*tsrv* and *tsmt*) files, this field is used to identify to the receiver of the file, the file name of the FileAct file.
ReferenceNumber	<Ref>	Max35Text	Transaction Reference Number.	For trade (*tsrv* and *tsmt*) files, this field is used to specify the reference number of the transaction to which the FileAct file is associated. If the reference number is not known, the value “NONREF” should be used.

XML Schema

```
<?xml version=“1.0“ encoding=“UTF-8“?>
<xs:schema xmlns=“urn:swift:xsd:ApplSpcfc.Trade.01“
xmlns:xs=“http://www.w3.org/2001/XMLSchema“
targetNamespace=“urn:swift:xsd:ApplSpcfc.Trade.01“ elementFormDefault=“qualified“>
<xs:element name=“ApplSpcfc“ type=“ApplSpcfc“/>
 <xs:complexType name=“ApplSpcfc“>
   <xs:sequence>
     <xs:element name=“FileHdrTrfTrad“ type=“FileHeaderTransferTradeV01“/>
   </xs :sequence>
 </xs :complexType>
 <xs :complexType name= »Document1 »>
   <xs:sequence>
     <xs:element name=”Nm” type=”Max70Text”/>
     <xs:element name=”Ref” type=”Max35Text”/>
   </xs:sequence>
 </xs:complexType>
 <xs:complexType name=”FileHeaderTransferTradeV01”>
   <xs:sequence>
     <xs:element name=”Doc” type=”Document1” minOccurs=”0”/>
   </xs:sequence>
```

```
  </xs:complexType>
  <xs:simpleType name="Max35Text">
    <xs:restriction base="xs:string">
      <xs:maxLength value="35"/>
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="Max70Text">
    <xs:restriction base="xs:string">
      <xs:maxLength value="70"/>
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Example

```
<ApplSpcfc xmlns = "urn:swift:xsd:ApplSpcfc.Trade.01" xmlns:xsi =
"http://www.w3.org/2001/XMLSchema-instance" >
   <FileHdrTrfTrad>
      <Doc>
         <Nm>SiemensScannedDocs.zip</Nm>
         <Ref>WQ667589A-20100519</Ref>
      </Doc>
   </FileHdrTrfTrad>
</ApplSpcfc>
```

2.7 Other Optional Fields

2.7.1 RequestRef field

The **RequestRef** field is an **optional** field that contains a unique user reference number for the file transfer assigned by the initiator of the operation.

2.7.2 File Description field

The **File Description** field is an **optional** field that enables users to enter additional non-structured information that is not meant for automated processing. Users are free to use this field to transfer any appropriate information describing the file content. However, SWIFT recommends the structured **FileInfo** field for this purpose.

ANNEX A: Format description codes

The below table gives the mandatory format description codes that have to be indicated in the request type field **when sending non-FIN or non-ISO 20022 formats through FileAct (e.g. local domestic formats)**.

The request type field is structured as <area>.<syntax_and_format>.<description>

Where **<area>** will be either

"pain" for payment initiation (debit or credit), or

"camt" for reporting standards.

<syntax_and_format> will be

"xxx."<format_description>,
with <format_description> one of the codes from the list below

<description> is optionally used for a further, bilaterally agreed specification

Note This table is updated progressively so users can always find the latest version on www.swift.com. If the format does not appear in the table, users must contact SWIFT so that the format can be added and published accordingly on swift.com.

Country Code	Published by	File format	<format description>
AU	Australian Payments Clearing Association	Fixed length 120 characters	becs
BE	Belgian Bankers' Association	Fixed length 128 characters	coda
BE	id.	Fixed length 128 characters	ciri
BR	Febraban CNAB	Fixed length 240 characters	cnab240
CH	Swiss Interbank Clearing	Fixed length 128 characters	dta
CH	id.	Variable length	lsvbdd
CH	PostFinance	Fixed length 128 characters	epo
CH	id.	Fixed length 128 characters	dd
DE	Zentraler Kreditausschuss	Variable length	dtaus
DE	id.	Variable Length	dtzav
DE	id.	Variable Length	dte
ES	Asociacion Espanola de Banca	Fixed length 72 characters	aeb34
ES	id.	Fixed Length 80 characters	aeb43
ES	Banco Santander Factoring	Fixed Length 170 characters	factor
ES	id.	Fixed length 162 characters	aeb19
ES	id.	Fixed length 162 characters	aeb58
FR	Comité Français d'Organisation et de Normalisation Bancaire	PLEASE CONSULT ANNEX B FOR FRANCE	
GB	APACS	Variable length	bacs
IE	IPSO – Irish Payment Services Organisation	Variable length	irishstd18
IE	id.	Variable length	irishstd27
IT	CBI-consortium (Customer to Business	Fixed length 120 characters	pc

Country Code	Published by	File format	<format description>
	Interaction)		
IT	id.	Fixed length 120 characters	pe
IT	id.	Fixed length 120 characters	ep
IT	id.	Fixed length 120 characters	ib
IT	id.	Fixed length 120 characters	ir
IT	id.	Fixed length 120 characters	im
IT	id.	Fixed length 120 characters	ab
IT	id.	Fixed length 120 characters	bb
IT	id.	Fixed length 120 characters	sl
IT	id.	Fixed length 120 characters	av
IT	id.	Fixed length 120 characters	ap
IT	id.	Fixed length 120 characters	ai
IT	id.	Fixed length 120 characters	al
IT	id.	Fixed length 120 characters	f24
IT	id.	Fixed length 120 characters	cn
IT	id.	Fixed length 120 characters	rh
IT	id.	Fixed length 120 characters	ra
IT	id.	Fixed length 120 characters	ec
IT	id.	Fixed length 120 characters	dt
IT	id.	Fixed length 120 characters	rp
IT	id.	*Credit transfer CBI* XML format *payment request*	ctx
IT	id.	*Credit transfer CBI* XML format *status report*	ctrix
IT	id.	*Credit transfer CBI* XML format *status report for the Debtor*	ctridx
IT	id.	*Credit transfer CBI* XML format *status report for the Creditor*	ctricx
IT	id.	*Sepa Direct Debit CBI XML format*	sddx
IT	id.	*Sepa Direct Debit CBI XML format Status report*	sddix
IT	id.	*Request for Invoice Advance CBI XML format request*	riax
IT	id.	*Request for Invoice Advance CBI XML format status report*	Riaix
IT	id.	*Structured Document*	desx

Country Code	Published by	File format	<format description>
		exchange CBI XML format	
IT	id.	*Non structured Document exchange CBI XML format*	densx
IT	id.	*Document exchange CBI XML format status report*	deix
IT	id.	*Domestic payments Banks transfer status report*	dptix
IT	id.	*Advice of incoming cross-border credit transfer*	aicbctx
IT	id.	*Advice of incoming cross-border credit transfer instructions*	aicbctinsx
JP	Japanese Bankers Association	Fixed length 120 characters	50200121
JP	id.	Fixed length 120 characters	50200111
JP	id.	Fixed length 120 characters	50200112
JP	id.	Fixed length 120 characters	50200191
JP	id.	Fixed length 120 characters	50200141
LU	Association des Banques et Banquiers, Luxembourg	Variable length	dom2000
LU	id.	Variable length	vir2000
NL	Interpay	Fixed length 50 characters	clieop02
NL	id.	Fixed length 50 characters	clieop03
NZ	New Zealand Bankers' Association	Fixed length 120 characters	bacho
PL	KIR (Polish National Clearing Chamber)	Variable length	PLI
PL	KIR (Polish National Clearing Chamber)	Variable length	PLA
PL	KIR (Polish National Clearing Chamber)	Variable length	STA
PT	Associaçao Portuguesa de Bancos	Fixed length 80 characters	ps2
US	ANSI	Variable length	820
US	BAI	Variable or fixed length	bai2
US	NACHA	Fixed length 94 characters	nacha
N/A	UN/CEFACT	EDIFACT	paymul96a
N/A	UN/CEFACT	EDIFACT	paymul95a
N/A	UN/CEFACT	EDIFACT	payord912
N/A	UN/CEFACT	EDIFACT	finsta96d
N/A	UN/CEFACT	EDIFACT	payext912
N/A	UN/CEFACT	EDIFACT	contrl96a

Country Code	Published by	File format	<format description>
N/A	UN/CEFACT	EDIFACT	bansta96a
N/A	UN/CEFACT	EDIFACT	dirdeb96a
N/A	UN/CEFACT	EDIFACT	cremul96a
N/A	UN/CEFACT	EDIFACT	debmul96a
N/A	UN/CEFACT	EDIFACT	creadv912
N/A	UN/CEFACT	EDIFACT	creext912
N/A	UN/CEFACT	EDIFACT	autack96a
N/A	UN/CEFACT	EDIFACT	cipher96a
N/A	UN/CEFACT	EDIFACT	author96a
N/A	SAP	IDOC	finsta01
N/A	SAP	IDOC	pexr2001
N/A	SAP	IDOC	pexr2002
N/A	SAP	IDOC	stats
N/A	Financial Insitution	Financial Institution specific	fisp
N/A	id.	ZIP file containing eStatement XML File Transfer Header and one or more PDF files of the statements (refer Annex C for further details)	estmt

ANNEX B: Country specifics

FRANCE

RequestTypes

France, as a community, has defined set of rules to create RequestTypes to be used between corporates and banks. These rules are common for all multi-bank channels and jointly maintained by CFONB and French SWIFT National User Group.

The **Request Type** field is mandatory and must be formatted as follows:

<area>.<syntax_and_format>.<description>

<area> code

<area> follows the common rules specified above.

<syntax_and_format>

<syntax_and_format> identifies the syntax and specific format used. This field is mandatory and must be formatted as specified above.

- `xxx.<format_description>` must be used either for CFONB defined formats and bank defined formats as described below :
  - for fixed length CFONB formats, “cfonb” followed by the file format length, (e.g. cfonb160),
  - for international formats used with French banks, even if CFONB documentation exists, the format name will follow the same rules listed above as in any other country (e.g. FIN, EDIFACT, ISO20022),
  - for bank proprietary formats, the format name given by the bank (for example vcom400 for VCOM with 400 characters fixed record length). Each bank will provide its customers with its proprietary list.

<description>

In addition to < syntax_and_format >, optional field <description > can be used to provide additional information on the content of the file. When used, this field is structured in two parts : <ddd>.<ppp>, both optional.

- <ddd> sub field to clearly identify the format when <format_description> is insufficient (e.g. cfonb240 which could be ACH return files or LCR payment validation). Values can be between the following list, in the common list above or bilaterally defined between bank and corporate.
- <ppp> defined bilaterally between a bank and its customer to give more precision about the file content, if needed.

<ddd> code	File content	File content (FR)	Direction
dct	Domestic credit transfer	Remise de virement domestique	C -> B
sct	SEPA credit transfer	Remise de virement SEPA	C -> B
ict	Intra group credit transfer (treasury transfer)	Remise de virement trésorerie France (Intra Groupe)	C -> B
xct	Cross-border credit transfer	Remise de virement international	C -> B
mct	Mixed credit transfer batch	Remise de virement	C -> B
rft	Request For Transfer	Remise de Request For Transfer	C -> B
ddd	Domestic direct debit	Remise de prélèvement domestique France	C -> B

dda	Accelerated Direct debit	Remise de prélèvement accéléré France	C -> B
dco	Domestic bills of exchange collection	Remise de LCR	C -> B
tch	Cheque batch data capture	Remise de Télécollecte chèques	C -> B
dvd	Check Account Identification Request	Demande vérification domiciliation	C -> B
dti	N/A	Remise de TIP	C -> B
dtg	N/A	Remise de Télérèglement	C -> B
sdd	SDD	Remise de SDD	C -> B
sbb	BtoB SDD	Remise de SDD BtoB	C -> B
bco	Bills of exchange acceptance	Bon à payer de LCR (Réponse au relevé de LCR)	C -> B
vct	Supplier domestic credit transfer	Remise de VCOM	C -> B
lch	Cheque letter remittance	Remise de lettre chèque	C -> B
ech	Cheques paid out remittance	Remise de chèque émis	C -> B
oth	Non specified	Remise d'opération	C -> B
pco	Bills of exchange statement	Relevé de LCR	B -> C
stm	End of period account statement	Relevé de compte	B -> C
rep	Intraday account statement	Relevé intraday	B -> C
dri	ACH info	Retour divers	B -> C
rct	List of rejected credit transfer	Relevé de virement rejeté	B -> C
rsd	List of rejected direct debit	Relevé de prélèvement rejeté	B -> C
rdd	List of rejected SDD	Relevé de SDD rejeté	B -> C
rbb	List of rejected B2B SDD	Relevé de SDD BtoB rejeté	B -> C
rco	List of unpayed Bills	Relevé de LCR impayé	B -> C
cai	Change Account Identification Request list	Relevé de changement de domiciliation	B -> C
ara	File acknowledgment	Accusé de Réception Applicatif	B -> C
ard	List of received direct debits	Relevé de prélèvement reçu	B -> C
add	List of received SDD	Relevé de SDD reçu	B -> C
abb	List of received B2B SDD	Relevé de SDD BtoB reçu	B -> C
act	List of received credit transfers	Relevé de virement reçu	B -> C
ati	N/A	Relevé de TIP reçu	B -> C
rti	N/A	Relevé de TIP rejeté	B -> C
atg	N/A	Relevé de Télérèglement reçu	B -> C
rtg	N/A	Relevé de Télérèglement rejeté	B -> C
ach	List of cheques to be paid	Relevé de chèque présenté au paiement	B -> C
rch	List of unpaid cheques	Relevé de chèque impayé	B -> C

Example list of Request Types

RequestType	Product description
pain.xxx.cfonb160.dct	Domestic credit transfer France Virement domestique France
pain.xxx.cfonb160.ict	Intra group credit transfer (treasury transfer) France Virement trésorerie domestique France
pain.xxx.cfonb160.ddd	Domestic direct debit Avis de prélèvement domestique France
pain.xxx.cfonb160.dco	Bill of exchange collection Remises de LCR
pain.xxx.cfonb240.bco	Bill of exchange acceptance Bon à payer de LCR (Réponse au relevé de LCR)
pain.xxx.cfonb320.dct	Domestic credit transfer France Virement domestique France
pain.xxx.cfonb320.xct	Cross border credit transfer Virement international
pain.xxx.cfonb320.rft	Request for Transfer Virement Request For Transfer
pain.xxx.400.vct	Supplier domestic credit transfer VCOM
camt.xxx.cfonb120.stm	End of period account statement Relevé de compte
camt.xxx.cfonb240.pco	Bills of exchange statement Relevé de LCR
camt.xxx.cfonb240.dri	ACH information Retour divers
camt.xxx.cfonb240.rct	List of rejected credit transfers Relevé de virements rejetés
camt.xxx.cfonb240.rsd	List of rejected direct debit Relevé de prélèvements rejetés
camt.xxx.cfonb240.rco	List of unpaid bill of exchange Relevé de LCR impayées
camt.xxx.cfonb240.cai	Change account identification request list Relevé de changement de domiciliation
camt.xxx.cfonb560.ara	File Control Acknowledgment Accusé de réception applicative

Request for Files specification

Note the below specification has been designed with the help of – and for – the banking community in France in the context of the replacement of the ETEBAC corporate to bank protocol; the approach described below is nevertheless generic and can be used in other contexts.

In some cases, the corporate may wish to initiate the reception of files from its banks, or need to be able to specify which files it wants to receive, e.g. all files created between 2 specific dates (in case previously received files were lost by accident.)

FileAct Download file mode allows initiating the reception of a file. In some cases however, this mode can not be used (e.g. sometimes it can take too long to prepare the file, leading to a time out of the download request). This section therefore addresses the above requirement by proposing a mechanism based on FileAct Transfer mode as pictured in fig 1:

1) The corporate sends a file (further referred to as the “RequestForFile”) containing the specifications of the file(s) requested from the bank,

2) The bank processes this RequestForFile and,

3) Sends back the requested file(s).

3’) In case the bank is unable to process the request (e.g. no files available), an exception file (further referenced as RequestForFileException) – containing the reason of rejection - is generated.

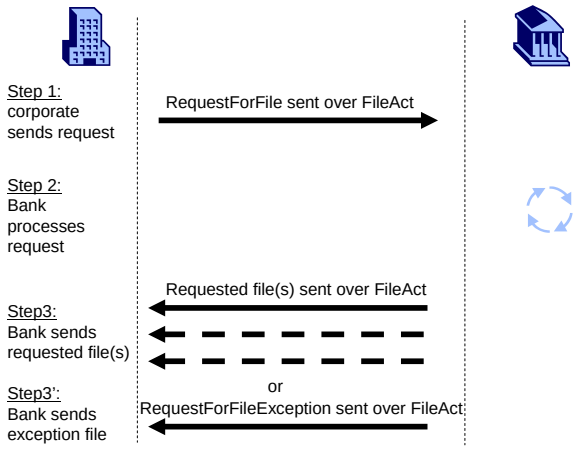


Fig 1.

This section specifies the contents of the RequestForFile and the RequestForFileException files, as well as the file transfer fields needed to send these files through FileAct;

File transfer fields

File transfer fields should be formatted in line with the specifications set out in section 3

Service name

as per section 2.2

Requestor and Responder DN

as per section 2.3

Request type field

This field should be formatted as follows:

For RequestForFile: xsys.xxx.requestforfile

For RequestForFileException: xsys.xxx.requestforfileexception

FileInfo field

as per section 2.5

Content of the RequestForFile

The RequestForFile should contain the following data using an xml structure

Element name	XML Tag	Occurrence	Format	Comment
QueryIdentification	QryId	Mandatory		
QueryReference	QryRef	Mandatory	Alphanumeric [1..35]	Reference of this request. This should have the same value as the UserRef field of the FileAct header.
Creation DateTime	CrDtTm	Mandatory	ISO Date time	Date time of the file request creation
FileQueryDefinition	FileQryDef	Mandatory		
FileType	FileTp	Mandatory	Alphanumeric [1..35]	Type of file; defined bilaterally (**note**: pre-defined set of keywords could be defined based on community proposal)
Criteria	Crt	Optional	Alphanumeric [1..140]	Criteria to be agreed upon bilaterally between corporate and bank; this field is expected to be filled in manually by an operator from the vendor's application

Content of the RequestForFileException

The RequestForFileException should contain the following data using an xml structure

Element name	Tag	Occurrence	Format	Comment
ExceptionIdentification	ExcptId	Mandatory		
ExceptionReference	QryRef	Mandatory	Alphanumeric [1..35]	Reference of this Exception. This should have the same value as the

				QueryReference appearing in the corresponding RequestForFile
Creation DateTime	CrDtTm	Mandatory	ISO Date time	Date time of the file Exception creation
ExceptionReason	ExcptRs	Mandatory	Alphanumeric [1..20]	Three possible values for this field: “NoFiles”: no files available “NoService”: customer has not subscribed to the service “CriteriaError”: the bank cannot interprete the specified criteria

Example 1: Request for files, with FileType = camt.xxx.cfonb120.stm

```
<RequestForFile>
  <QryId>
    <QryRef>Sample1</QryRef>
    <CrDtTm>2009-01-11T11:13:32+02:00</CrDtTm>
  </QryId>
  <FileQryDef>
    <FileTp> camt.xxx.cfonb120.stm </FileTp>
  </FileQryDef>
</RequestForFile>
```

Example 2: Request for specific files at a specific date :

```
<RequestForFile>
  <QryId>
    <QryRef>Sample3</QryRef>
    <CrDtTm>2009-01-11T11:31:22+02:00</CrDtTm>
  </QryId>
  <FileQryDef>
    <FileTp>camt.xxx</FileTp>
    <Crt>Date=2009-01-10</Crt>
  </FileQryDef>
</RequestForFile>
```

Example 3: reject of the previous request (answer is NoFiles) :

```
<RequestForFileException>
  <ExcptId>
    <QryRef>Sample3</QryRef>
    <CrDtTm>2009-01-11T11:31:22+02:00</CrDtTm>
  </ExcptId>
  <ExcptRs>NoFiles<ExcptRs>
</RequestForFileException>
```

XSD Request For Files

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema elementFormDefault="qualified" xmlns="urn:swift:corp:xsd:corp.1.0"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:swift:corp:xsd:corp.1.0">
    <xs:simpleType name="Max35Text">
        <xs:restriction base="xs:string">
            <xs:minLength value="1" />
            <xs:maxLength value="35" />
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="Max140Text">
        <xs:restriction base="xs:string">
            <xs:minLength value="1" />
            <xs:maxLength value="140" />
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="QryId">
        <xs:sequence>
            <xs:element name="QryRef" type="Max35Text" />
            <xs:element name="CrDtTm" type="ISODateTime" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="FileQryDef">
        <xs:sequence>
            <xs:element name="FileTp" type="Max35Text" minOccurs="1"
                maxOccurs="1" />
            <xs:element name="Crt" type="Max140Text" minOccurs="0"
                maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="RequestForFile">
        <xs:sequence>
            <xs:element name="QryId" type="QryId" minOccurs="1"
                maxOccurs="1" />
            <xs:element name="FileQryDef" type="FileQryDef" minOccurs="1"
                maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
    <!-- Root element -->
    <xs:element name="RequestForFile" type="RequestForFile" />
</xs:schema>
```

XSD Request For Files Exception

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema elementFormDefault="qualified" xmlns="urn:swift:corp:xsd:corp.1.0"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:swift:corp:xsd:corp.1.0">
    <xs:simpleType name="Max20Text">
        <xs:restriction base="xs:string">
            <xs:minLength value="1" />
            <xs:maxLength value="20" />
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="Max35Text">
        <xs:restriction base="xs:string">
            <xs:minLength value="1" />
            <xs:maxLength value="35" />
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="QryId">
        <xs:sequence>
            <xs:element name="QryRef" type="Max35Text" />
            <xs:element name="CrDtTm" type="ISODateTime" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="RequestForFileException">
        <xs:sequence>
            <xs:element name="ExcptId" type="QryId" minOccurs="1"
                maxOccurs="1" />
            <xs:element name="ExcptRs" type="Max20Text" minOccurs="1"
                maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
    <!-- Root element -->
    <xs:element name="RequestForFileException" type="RequestForFileException"
        />
</xs:schema>
```

ANNEX C: eStatements Supplementary Information

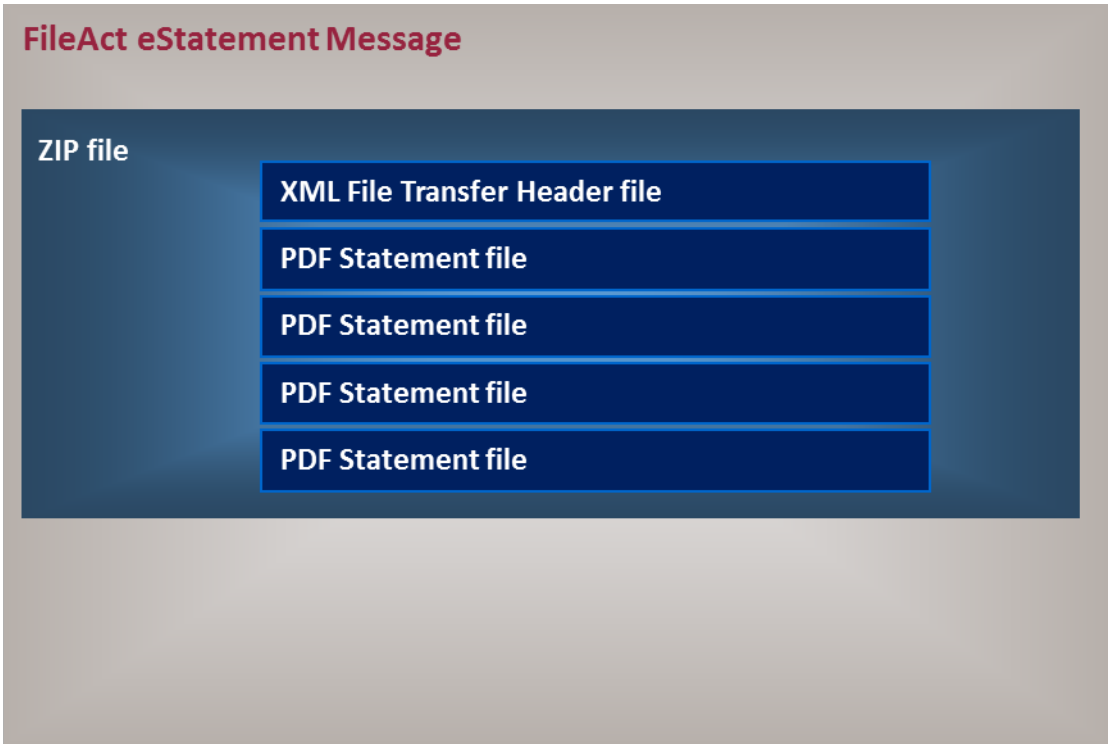
An eStatement is an electronic copy of a corporation's period end statement in a PDF (Adobe Portable Document Format) file that is transmitted over SWIFT FileAct to the account owner. eStatements allow financial institutions to securely transmit period end, legal bank statements, eliminating the need for paper statements, transportation, mailing and storage.

eStatements are used by a corporation's internal teams in treasury that need the ability to view, research and print the actual legal statements for audit, control and compliance purposes. There is a distinction between an eStatement versus the daily files received by a corporation in MT940/942/BAI2/BTRS/camt format(s) used for liquidity management and cash reconciliation.

eStatements are generated by a financial institution as one or more PDF files that are then collected and compressed as a single ZIP file for transmission. In addition, each ZIP file contains an XML File Transfer Header file which provides further information on the contents of the ZIP file including for each statement:

- Bank BIC
- Account Number
- Statement Period Start Date
- Statement Period End Date
- Statement Type

The following sections depict the structure of the FileAct message and detail the format of the **XML File Transfer Header** file.



The **XML File Transfer Header** file is formatted as an XML structure, the data elements are detailed in the following table:

Element Name	XML Tag	Presence	Data Type	Description
File Name	<FileNm>	[0..1]	Max256 Text	Name of the compressed file in case multiple files are present within the FileAct envelope. Name of the ZIP file.
Count Of Files	<CntOfFiles>	[1..1]	Max15 Numeric Text	Number of files contained the compressed file in case multiple files are present within the FileAct envelope. Count of the number PDF files in the ZIP file.
File Description	<FileDesc>	[1..*]		Repeating Component, containing below indented elements
File Name	<FileNm>	[1..1]	Max256 Text	Name of the file in FileAct envelope. Name of individual PDF statement file.
Statement Type	<StmtTp>	[0..1]	Max4 AlphaNumeric Text	Type of electronic statement that is provided in the file. Bilaterally determined.
Account Servicer	<AcctSvcr>	[0..1]	BIC8 or BIC11	Identification of the financial institution servicing the account who has generated the electronic statement. BIC code.
Account Identification	<AcctId>	[1..1]	Max34 Text	Identification of the account for which the electronic statement is generated. Either IBAN or Other
Start Date	<StartDt>	[1..1]	ISODate YYYY-MM-DD	Date at which the file starts. Statement start date.
End Date	<EndDt>	[1..1]	ISODate YYYY-MM-DD	Date at which the file ends. Statement start date.

XML File Transfer Header Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:swift:xsd:ApplSpcfc.eStmtV02"
xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:swift:xsd:ApplSpcfc.eStmtV02"
elementFormDefault="qualified">
    <xs:element name="ApplSpcfc" type="ApplSpcfc"/>
    <xs:complexType name="ApplSpcfc">
     <xs:sequence>
        <xs:element name="ElctrncStmtFileTrfHdr" type="ElectronicStatementFileTransferHeaderV02"/>
     </xs:sequence>
    </xs:complexType>
    <xs:complexType name="AccountIdentification4Choice">
     <xs:sequence>
        <xs:choice>
          <xs:element name="IBAN" type="IBAN2007Identifier"/>
```

```
            <xs:element name="Othr" type="GenericAccountIdentification1"/>
          </xs:choice>
       </xs:sequence>
      </xs:complexType>
      <xs:complexType name="AccountSchemeName1Choice">
       <xs:sequence>
          <xs:choice>
            <xs:element name="Cd" type="ExternalAccountIdentification1Code"/>
            <xs:element name="Prtry" type="Max35Text"/>
          </xs:choice>
       </xs:sequence>
      </xs:complexType>
      <xs:simpleType name="AnyBICIdentifier">
       <xs:restriction base="xs:string">
          <xs:pattern value="[A-Z]{6,6}[A-Z2-9][A-NP-Z0-9]([A-Z0-9]{3,3}){0,1}"/>
       </xs:restriction>
      </xs:simpleType>
      <xs:complexType name="ElectronicStatementFileDescription2">
       <xs:sequence>
          <xs:element name="FileNm" type="Max256Text"/>
          <xs:element name="StmtTp" type="Max4AlphaNumericText" minOccurs="0" maxOccurs="1"/>
          <xs:element name="AcctSvcr" type="AnyBICIdentifier" minOccurs="0" maxOccurs="1"/>
          <xs:element name="AcctId" type="AccountIdentification4Choice"/>
          <xs:element name="StartDt" type="ISODate"/>
          <xs:element name="EndDt" type="ISODate"/>
       </xs:sequence>
      </xs:complexType>
      <xs:complexType name="ElectronicStatementFileTransferHeaderV02">
       <xs:sequence>
          <xs:element name="FileNm" type="Max256Text" minOccurs="0" maxOccurs="1"/>
          <xs:element name="CntOfFiles" type="Max15NumericText"/>
          <xs:element name="FileDesc" type="ElectronicStatementFileDescription2" minOccurs="1"
maxOccurs="unbounded"/>
       </xs:sequence>
      </xs:complexType>
      <xs:simpleType name="ExternalAccountIdentification1Code">
       <xs:restriction base="xs:string">
          <xs:minLength value="1"/>
          <xs:maxLength value="4"/>
       </xs:restriction>
      </xs:simpleType>
      <xs:complexType name="GenericAccountIdentification1">
       <xs:sequence>
          <xs:element name="Id" type="Max34Text"/>
          <xs:element name="SchmeNm" type="AccountSchemeName1Choice" minOccurs="0"
maxOccurs="1"/>
          <xs:element name="Issr" type="Max35Text" minOccurs="0" maxOccurs="1"/>
       </xs:sequence>
      </xs:complexType>
      <xs:simpleType name="IBAN2007Identifier">
       <xs:restriction base="xs:string">
          <xs:pattern value="[A-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{1,30}"/>
       </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="ISODate">
       <xs:restriction base="xs:date"/>
```

```
    </xs:simpleType>
    <xs:simpleType name="Max15NumericText">
     <xs:restriction base="xs:string">
        <xs:pattern value="[0-9]{1,15}"/>
     </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="Max256Text">
     <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="256"/>
     </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="Max34Text">
     <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="34"/>
     </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="Max35Text">
     <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="35"/>
     </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="Max4AlphaNumericText">
     <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="4"/>
        <xs:pattern value="[a-zA-Z0-9]{1,4}"/>
     </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

XML File Transfer Header Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ApplSpcfc xsi:schemaLocation="urn:swift:xsd:ApplSpcfc.eStmtV02 ApplSpcfc.eStmtV02.xsd"
xmlns="urn:swift:xsd:ApplSpcfc.eStmtV02" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ElctrncStmtFileTrfHdr>
        <FileNm>WASH20131021000671A</FileNm>
        <CntOfFiles>2</CntOfFiles>
        <FileDesc>
            <FileNm>EODACC1_20131020</FileNm>
            <StmtTp>EOD</StmtTp>
            <AcctSvcr>HSBCBNBB</AcctSvcr>
            <AcctId>
                <Othr><Id>111-222222-666</Id></Othr>
            </AcctId>
            <StartDt>2013-10-20</StartDt>
            <EndDt>2013-10-20</EndDt>
        </FileDesc>
        <FileDesc>
            <FileNm>EODACC2_20131020</FileNm>
            <StmtTp>EOD</StmtTp>
            <AcctSvcr>HSBCBNBB</AcctSvcr>
```

```
            <AcctId>
                <Othr><Id>111-222222-777</Id></Othr>
            </AcctId>
            <StartDt>2013-10-20</StartDt>
            <EndDt>2013-10-20</EndDt>
        </FileDesc>
    </ElctrncStmtFileTrfHdr>
</ApplSpcfc>
```