

eBAM



File Naming Convention for attachments to EBAM messages

This document describes the rules that users must follow to name files related to eBAM messages, exchanged using FileAct.

July 2010



Preface

Purpose of this document

This document describes the rules that users must follow when naming attachments related to eBAM messages exchanged over FileAct. For corporate to bank communication, this naming convention will be imposed in SCORE (Standardised CORporate Environment).

This version 0.1 (June 2010), is a first draft proposal to be validated by a limited number of early adopters of the eBAM solution on SWIFT.

Intended audience

BNY Mellon

Related documentation

The following documents contain information related to this naming convention proposal:

- [SWIFT for corporates – FileAct Implementation Guide](#) – chapters 2.4.2, 2.5.2 and 2.6.2 detail the use of the FileAct header and the way to attach related documents to an eBAM message in the context of SCORE.

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
<i>Italics</i>	Important information and document names
Courier	User input, directory paths, parameter values, place holders, and system output examples

Table of Contents

1	Overview	4
1.1	Background.....	4
1.2	Requirements.....	4
1.3	Restrictions.....	4
2	File naming convention	5
2.1	“TypeOfFile” element.....	5
2.2	“NameOfFile” element.....	5
2.3	Examples:.....	6
ANNEX A:	FileAct header information for BAM	7
A.1	Request Type field.....	7
A.2	FileInfo field.....	7
A.3	HeaderInfo field.....	8

1 Overview

1.1 Background

When exchanging Bank Account Management (BAM) information, it is necessary to support the instructions with additional documentation such as scans of passports, the bank's terms and conditions, the corporate's by-laws, etc... The types of documents that are required for a specific bank account management function, and their formats, are often bank policy dependent and/or regulatory defined.

In order to easily and automatically process this additional information in a straight through manner, it is important to clearly and uniformly identify the attachments. Therefore, this document is providing a naming convention that, at least in the scope of SCORE (Standardised CORporate Environment), will be mandated.

For eBAM information exchanged over FileAct, there is a description on how to identify whether and which attachments are exchanged. The FileAct Implementation Guide describes the use of an XML schema in the HeaderInfo field of FileAct header for this purpose (extract in Annex A). The XML schema allows for unique identification of each attached file, with the type of the file and its name.

1.2 Requirements

- A list of codes, as exhaustive as possible, needs to identify the file types, exchanged as attachments of an EBAM message,
- The identification of the attachment needs to uniquely link it to its sender.
- The identification of the attachment needs to allow for detection of potential duplicate attachments,
- The identification of the attachment needs to be independent from the EBAM XML message (to allow for separate submissions)
- The identification of the attachment needs to be independent from the FileAct exchange (to allow for different ways to transport the file to the bank)
- The naming convention applies to attachments to all EBAM messages, irrespective of the direction of the flow (bank to corporate, corporate to bank, ...)

1.3 Restrictions

- The "TypeOfFile" element in the XML schema is restricted to 35 characters
- The "NameOfFile" element in the XML schema is restricted to 70 characters.

2 File naming convention

2.1 “TypeOfFile” element

The following codes can be used to identify the type of document that is attached to the EBAM message:

Code	Meaning
TECO	Terms & conditions
PASS	Passport
IDCA	Identity card
DRLI	Driver’s License
SOSE	Social Security Card
BYLA	By-laws, Articles of Association, Statuts
MASA	Master Agreement
SERA	Service Agreement
CORE	Corporate Resolutions
SICA	(Wet) Signature Card
OTHR	Other

2.2 “NameOfFile” element

The **NameOfFile** element allows for unique identification of the attachment(s) by using the following structure:

<BIC-8>_<timestamp>_<sequencenbr>_<content>.<format>[.<zip>]

- **<BIC-8>** mandatory element which identifies the sender of the attachment. The format is `BBBBCCZZ`
- “_” (underscore)
- **<timestamp>** mandatory element which identifies date and time of the exchange. Also allows for identifying a new version of a similar document. The format is `YYYYMMDDHHMMSS`. The time is local to the creator of the message.
- “_” (underscore)
- **<sequencenbr>** mandatory element which allows to distinguish multiple attachments from a single sender in one exchange. The format is `NNNN`
- “_” (underscore)
- **<content>** mandatory element which further identifies the actual content of the file, beyond the identification done by the element “TypeOfFile”. The format is `max31alphanumeric`, No dot (“.”) allowed.
- “.” (dot)
- **<format>** mandatory element to identify the format of the file according to the commonly used file extensions. The format is `xxx[x]`. Individual institutions may agree with their counterparties on a restriction of the formats that are accepted as part of an EBAM attachment.
- **[“.” <zip>]** optional element to identify whether and which compression algorithm has been used on the file. The format is `xxx[x]`. Individual institutions may agree with their counterparties on a restriction of the compression algorithms that are accepted as part of an EBAM attachment.

In an EBAM file exchanged over FileAct, the value of this NameOfFile element in the HeaderInfo field in the FileAct header must be equal to the value of "AttachedDocumentName" (item 1.3) of the EBAM XML message contained in the same FileAct file:

- acmt.007.001.01 (AccountOpeningRequestV01)
- acmt.008.001.01 (AccountOpeningAmendmentRequestV01)
- acmt.013.001.01 (AccountReportRequestV01)
- acmt.015.001.01 (AccountExcludedMandateMaintenanceRequestV01)
- acmt.016.001.01 (AccountExcludedMandateMaintenanceAmendmentRequestV01)
- acmt.017.001.01 (AccountMandateMaintenanceRequestV01)
- acmt.018.001.01 (AccountMandateMaintenanceAmendmentRequestV01)
- acmt.019.001.01 (AccountClosingRequestV01)
- acmt.020.001.01 (AccountClosingAmendmentRequestV01)

If attachments are added to the returning EBAM XML messages, their names may also be present in "AttachedDocumentName" (item 1.6) of the:

- acmt.009.001.01 (AccountOpeningAdditionalInformationRequestV01)
- acmt.010.001.01 (AccountRequestAcknowledgementV01)
- acmt.011.001.01 (AccountRequestRejectionV01)
- acmt.012.001.01 (AccountAdditionalInformationRequestV01)
- acmt.014.001.01 (AccountReportV01)
- acmt.021.001.01 (AccountClosingAdditionalInformationRequestV01)

2.3 Examples:

Type Of File element	Name Of File element	Content of file
PASS	GENEUS33_20100715124533_5233_JOHN SMITH.jpeg	A jpeg-formatted scan of the passport of John Smith, created by General Electric on 15 July 2010, at 12:45:33 and sent as 5233 rd document
BYLA	MSFTUS66_20100715093445_0001_Venezuela subsidiary.pdf	An Acrobat formatted document that contains the by-laws for the Venezuelan subsidiary of MicroSoft, created by them on 15 July 2010, at 9:34:45

ANNEX A: FileAct header information for BAM

A.1 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.

<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

A.2 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

The **FileInfo** field is **mandatory** for BAM-files 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

A.3 HeaderInfo field

The **HeaderInfo** field can be used to specify processing instructions when sending information through FileAct.

The **HeaderInfo** field may contain, for information purposes, the name of the attachments - as per naming convention detailed in this document- 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 type 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-type of an attachment, using the pre-defined list of codes in chapter 2.1 of this document.
NameOfFile	<Nm>	Max70Text	Name 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 filename of an attachment, using the file naming convention explained in chapter 2.2 of this document..

XML Schema

```
<?xml version = "1.0" encoding = "UTF-8"?>
<xs:schema xmlns="urn:swift:xsd:ApplSpfcf.BAM.01"
targetNamespace="urn:swift:xsd:ApplSpfcf.BAM.01"
xmlns:xs=http://www.w3.org/2001/XMLSchema elementFormDefault="qualified">
<xs:element name = "ApplSpfcf" type = "ApplSpfcf"/>
<xs:complexType name = "ApplSpfcf">
<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>TECO</Tp>
      <Nm> CORPCCZZ_20100715124533_5233_terms for Venezuela subsidiary.pdf </Nm>
    </File>
    <File>
      <Tp>PASS</Tp>
      <Nm> CORPCCZZ_20100715083532_0001_John Smith.jpeg </Nm>
    </File>
  </FileTrfHdrBAM>
</ApplSpcfc>

```