

**USER GUIDE FOR THE OECD STANDARD TRANSMISSION
FORMAT**

**The OECD Standard Transmission Format
for international information exchange in taxation**

An introduction

Content

1. Where does STF fit in?
2. What content is STF supposed to support?
3. What is the main structure of an STF message?
4. What is the modular structure of the schemas for STF definition?
5. What is the structure of an STF_DIRECT document?
6. Where is detailed advice for all of the elements and their content to be found?
7. How is coexistence between SMF and STF guaranteed?
8. Examples of Elements and Messages
9. What artefacts are available for STF?

1. Where does STF fit in?

1.1 STF has been defined as the successor of SMF, the OECD Standard Magnetic Format for international information exchange in direct taxation, adopted in 1992 and re-formulated in 1997. To date there is no time limit set for the co-existence of the SMF with the STF.

1.2 STF is part of the SEIT (Standards for Exchange of Information in Taxation) family of OECD recommendations for international information exchange in taxation. In this set of recommendations the responsibility of STF is to define the format of message content, which is achieved by way of Extensible Markup Language (XML) schema. STF is not concerned with the way messages are transmitted, encrypted etc.

1.3 Whilst an important design objective for STF was to stay as closely as possible with SMF (thus making bridging programs possible), it is also a medium term goal to make STF compatible with emerging international XML standards in taxation as aspired to by the Organization for the Advancement of Structured Information Standards (OASIS) TaxXML Technical Committee (TC). It is the intention both of the OASIS TaxXML TC and the OECD TIES group to work for such convergence.

2. What content is STF supposed to support?

2.1 SMF was constructed to support automatic information exchange (in the sense of Article 26 of the OECD Model Convention) for direct tax purposes. Being primarily – even if not only - a modern version of SMF, STF, too, supports this kind of exchange. So the first message format built with STF has been STF_DIRECT for exactly this sort of information.

2.2 It was, however, also a design objective for STF to be flexible and extensible. Therefore STF can easily be extended for any other kind of tax information messages. This includes both the use for other than automatic exchange and for other content than the conventional income information of the SMF type.

3. What is the main structure of an STF message?

3.1 As usual for messages, STF messages are hierarchically structured with a header (MessageSpec) specifying technical information for the message as a whole and an arbitrary number of detail documents. (In a context like this we use the word “document” in a general sense, not in the strict meaning of XML, where a document is always the most comprehensive unit that contains one and only one root element. Documents in the strict XML sense are what we call messages here.) In the present state of STF development there is only one kind of such documents defined (STF_DIRECT), but as soon as other document formats will be developed, they can be included in such messages as well.

Figure 1 depicts this overall structure.

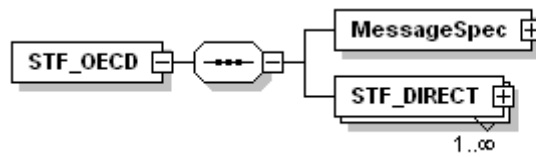


Figure 1: Overall Structure of STF Messages

In XML schema terms this is expressed as

```

<xsd:element name="STF_OECD">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="MessageSpec" type="MessageSpec_Type"/>
      <xsd:element name="STF_DIRECT" type="STF_Direct_Type" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="version" fixed="1.0"/>
  </xsd:complexType>
</xsd:element>

```

SchemaFragment 1

An attribute of name “version” and value “1.0” designates the current status of development.

3.2 The structure of the message specification (header) element is shown in figure 2.

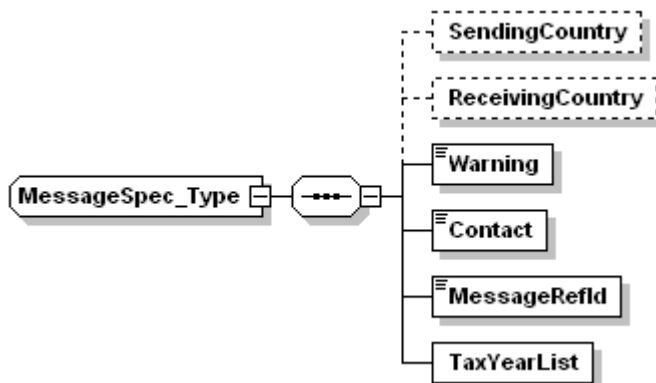


Figure 2: Structure of the Message Specification Header

The element contains data identifying and describing the message as a whole.

'SendingCountry' and 'ReceivingCountry' are to identify the relation of the transmission, so this is visible at the very top of the message and independent of the transmission content further downstream. The elements are optional because

- they are not indispensable for successful transmission, ,
- there are no exactly corresponding fields in the SMF record
- STF shall closely resemble the SMF.

It is, however, strongly recommended to use these identifying fields as in tended.

'Warning' is for legal constraints: free text expressing the restrictions for use of the information this message contains and the legal framework under which it is exchanged.

'Contact' should contain all necessary contact information about persons responsible for and involved in the processing of the data transmitted in this message, both legally (competent authority) and technically. This is free text as it is not intended for automatic processing.

'MessageRefId' is a unique identifier that the sender has to attribute to this message and shall

be used in any correspondence. 'TaxYearList' is a list of all tax years for which information is transmitted in the documents of the current message. To indicate a tax year, the date of the last day of that year is given. Format for dates is ccyy-mm-dd. List items have to be separated by blanks.

4. What is the modular structure of the schemas for STF definition?

4.1 STF documents are XML documents. The TIES Technology Task Team has defined:

(1) an XML schema document (stftypes-1.0.xsd) containing a set of simple and complex data types for the use in any STF schema defining a particular document type

(2) an XML schema document (stfdirect-1.0.xsd) for the definition of the XML documents that will replace SMF records, together with the definition of the message container STF_OECD for these documents

(3) two additional XML schema documents for OECD and ISO code lists to be used in STF documents, these schemas contain enumerations of the admissible code-values.

4.2 The core of STF is the definition of the data types to be used in STF documents. It is expected that this set of types will be extended as soon as new documents will be defined. With the advent of new document definitions there will certainly arise additional needs that were not yet addressed from a purely stfdirect perspective. Such new types are expected to fall into three categories:

(1) types that – even if not necessary for stfdirect – are of a certain generality and shall therefore be added to the stftypes collection

(2) types that are specially needed for just a certain document definition without a more general usefulness; they shall be added in a separate XML schema for the use of that one document definition only

(3) types that though close to others already defined in stftypes differ somewhat for the modelling of some aspect in the new document; they shall be derived as extensions or restrictions of their general stftypes relatives.

4.3 As long as stfdirect is the only document type in the STF family, it is considered adequate not to complicate the schema structure more than needed for this situation. Therefore:

(1) the general message structure and the stfdirect document structure are defined inside the same schema;

(2) all the above mentioned schemas are put into the same namespace (urn:oe.cd:ties:stf:v1);

This results in the structure shown in figure 3.

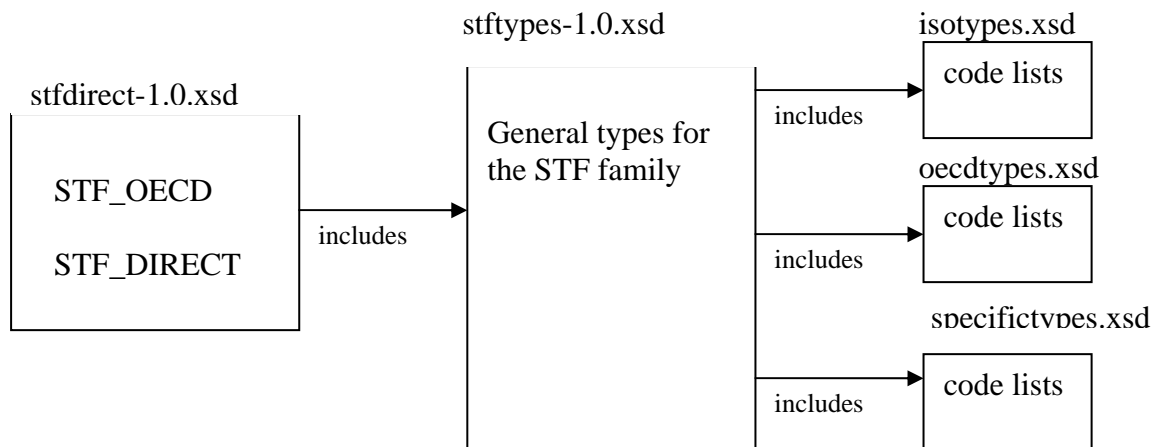


Figure 3: Inclusion Structure of STF schemas (to date)

4.4 When new document types are added in the future, the structure of figure 3 will probably not be adequate any more. Every document type will then be defined in a separate namespace together with its special data types and the general STF (core) types including the type for the message will be imported.

5. What is the structure of an STF_DIRECT document?

5.1 The high-level structure of an STF_DIRECT document is shown in figure 4.

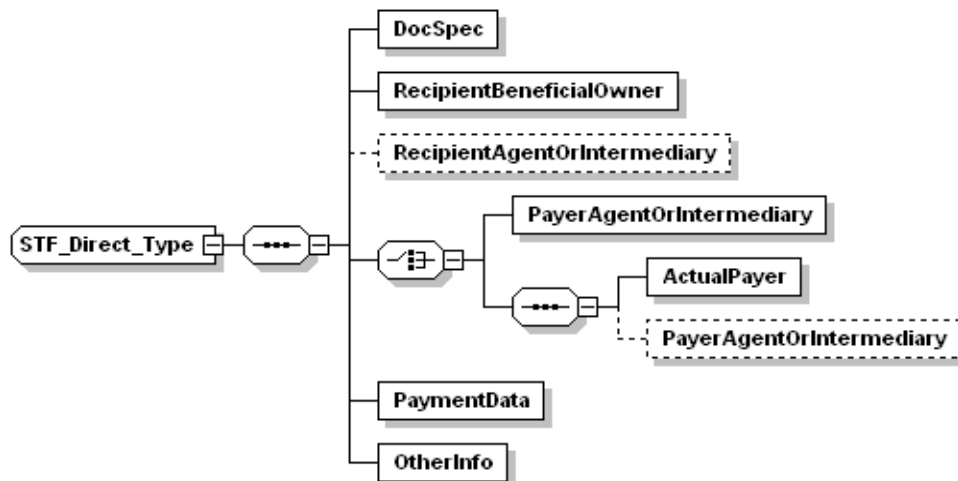



Figure 4: STF_DIRECT, highest level

It will be noted that the components of this structure fall into four categories:

- DocSpec, PaymentData and OtherInfo each represent a particular type of information occurring once in the document.
- All other components are of the same category: they denote parties in the transaction reported.

The construction may at first seem somewhat complicated, but should be rather clear after some inspection. A dotted line box indicates “optional”, data for parties so denoted can either be present or not, boxes with solid lines indicate obligatory entries. In a document of stfdirect type data for the beneficial owner must be present, whereas data for an agent or intermediary acting on behalf of the beneficial owner need not be present. The modelling of the data for the payer side is a bit more sophisticated. We have to make sure that data for at least one of “actual payer” and “payer agent or intermediary” are provided, but there is no

stringent rule that a particular one of those is obligatory. The “choice” symbol  stands for “one of these”. You may want to verify that the construction given in Figure 4 allows for all of these situations:

- actual payer data only
- payer agent or intermediary data only
- both actual payer and payer agent or intermediary data

and at the same time requires one of these situations .

5.2 The DocSpec element serves as a descriptor of the particular stfdirect document to which it belongs, just as the MessageSpec element does for the whole collection of documents in the message. DocSpec has this structure:

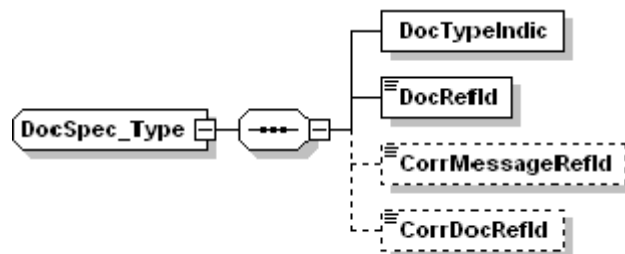


Figure 5: Document Specification Element

The document type indicator (DocTypeIndic) contains administrative data about the status of the document (is it “new” data sent for the first time – the normal case, hopefully near to 100% of documents -, or is it a correction of a document sent before, or is it a repetition of a document which was sent before but possibly not received in an orderly way).

The document reference identification (DocRefId) is the unique identifier of this document. For later reference to be possible it has to be unique at least within the message in which it is contained.

The following two elements are optional and only needed in case of a correction or a repetitive sending. (As they are actually obligatory and not optional in these cases, the schema here is a somewhat weak model of the overall situation.) The elements refer to documents

sent before by giving the DocRefId and MessageRefId of the document referred to and the message it was in.

5.3 Payment data are the reason why the document is sent. Here the sending administration enters the information that has become known about income of the beneficial owner in the source country. Here is the structure of the element:

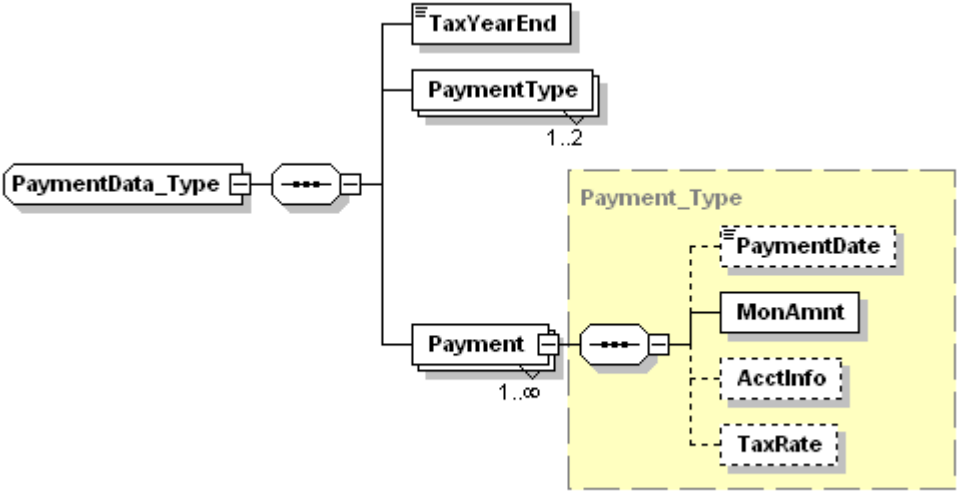


Figure 6: Data about the Income

Each single document serves for information about one and only one income item of the beneficial owner. It follows that several documents have to be transmitted (preferably in the same message) if there is the need to inform about income from several sources, at several points of time etc.

The tax year to which the payment belongs is entered in the element TaxYearEnd, which is a date field (format ccyy-mm-dd in coherence with general XML rules). This is not just a four digit element for the year in order to cope with cases where the tax year does not coincide with the calendar year.

The type of the payment received by the beneficiary is coded in the elements OECDPaymentType and SpecificPaymentType. Their structure is governed by these schema definitions:

```

<xsd:simpleType name="OECDPaymentType_Type">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      The OECD code describing the nature of the payments:
      06 Income from immovable property
      .....
      21 Other income
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="06"/>
    .....
    <xsd:enumeration value="21"/>
  </xsd:restriction>
</xsd:simpleType>

```

```

<xsd:complexType name="SpecificPaymentType_Type">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">Type for explanation of a payment by a code that is specific for a certain
legislation, e.g. for the sending country. In the OECD file for this schema part is a dummy code. The enumeration
element and the annotation-documentation in the OECD prepared file serve as an example for real legislation specific
codes and their documentation.</xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="SpecifPaymentType_Type">
      <xsd:attribute name="specificPaymentTypeQlf" type="xsd:string" fixed="Dummy"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="SpecifPaymentType_Type">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="00"/>
  </xsd:restriction>
</xsd:simpleType>

```

SchemaFragment 2b

In order to provide sufficient freedom for describing the nature of the income a country/legislation specific payment code may be included in addition to the standard OECD payment code. A sending country may want to transmit a special income code used in this country to best describe what income the beneficiary has received.

If no such specific code is transmitted, the element `SpecificPaymentType` should not be used. If it is used nevertheless, it has to look exactly like this in order to keep the document from becoming invalid:

```
<SpecificPaymentType specificPaymentTypeQlf="Dummy">00</SpecificPaymentType>
```

A sending country that wants to use specific payment codes has to edit the file `specifictypes_v1.xsd`. This file keeps the (country) specific codes (by now just for payment types) separate from the general OECD types. The attribute `specificPaymentTypeQlf`, which has to be fixed for all documents sent by this country and relying on this particular set of payment codes, has to be set to a value identifying this code list (e.g. country code + year). Then the enumeration of the codes in `specifictypes_v1.xsd` has to be adjusted according to need and should be accompanied by proper explanation of the meaning of the codes.

For the payment itself there is a multiplicity of elements. It has to be borne in mind that all of these elements belong to one and only one income item; they represent different aspects of this income item, as the gross payment, the net payment, the tax, and the refund. Here is what the element has to look like in XML schema format:

```

<xsd:complexType name="Payment_Type">
  <xsd:sequence>
    <xsd:element name="PaymentDate" type="xsd:date" minOccurs="0"/>
    <xsd:element name="MonAmnt" type="MonAmnt_Type"/>
    <xsd:element name="AcctInfo" type="AcctInfo_Type" minOccurs="0"/>
    <xsd:element name="TaxRate" type="TaxRate_Type" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="paymentQlf" type="paymentQlf_Type" use="required"/>
</xsd:complexType>

```

SchemaFragment 3

The payment qualifier (paymentQlfl) is the attribute which distinguishes gross, net etc. and has to be one of gip (gross income paid), nip (net income paid), twh (tax withheld), and trf (tax refunded). The tax rate (TaxRate) can optionally be given for any payment item. Tax rates have to be entered as decimal numbers with a total of four digits, two before and two after the decimal point. The date of the payment can be added to any of the items and should designate the day specific to the particular payment, e.g. the refund. Monetary amounts in STF are always qualified by an attribute currCode which is to give the ISO 4217 currency code relevant for the number in the MonAmnt element. For cases where the account into which the payment went matters (and is known) there is a field AcctInfo available, which looks like this:

```
<xsd:complexType name="AcctInfo_Type">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:choice>
      <xsd:element name="IBAN" type="IBAN_Type"/>
      <xsd:element name="OBAN" type="OBAN_Type"/>
      <xsd:element name="ISIN" type="ISIN_Type"/>
      <xsd:element name="OSIN" type="OSIN_Type"/>
      <xsd:element name="SWIFT" type="SWIFT_Type"/>
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>
```

SchemaFragment 4

The IBAN, ISIN, and SWIFT elements shall contain account identifiers as their names say and shall have the standard format of the respective identifiers. OBAN and OSIN stand for “other bank account number” and “other securities identification number” and are to be used for non-standard cases; attributes 'acctNoQlfl' and 'secNoQlfl' respectively have to be used to indicate the kind of such numbers.

5.4 Other information that may be needed to adequately describe the case at hand isn't part of the element PaymentData but goes into the element OtherInfo. There are no restrictions to the format of this element, which may also have child elements. The sender has to make sure both by using adequate tag names and adding explanations that the receiver is able to understand the sender's intention. As the document is possibly processed automatically there is no guarantee when or even that the content will be recognized by the receiver.

5.5 Identification of the parties involved in the payment is vital for the document to be of any value at all. Therefore a large part of the document content is given to data describing the parties. This is done in a uniform way for all parties, in XML language: all party elements like RecipientBeneficialOwner, ActualPayer etc. are of the same type, Party_Type. So we will have to get acquainted with Party_Type. Here is the broad picture:

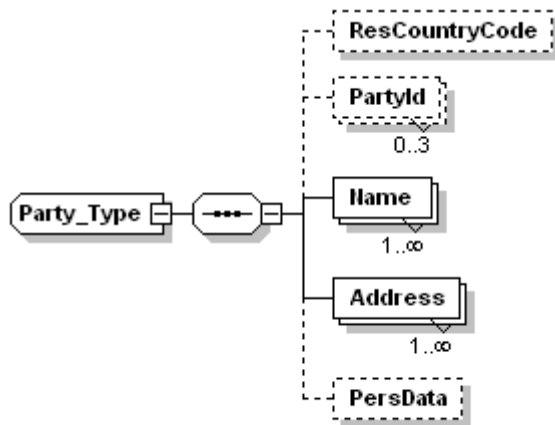


Figure 7: Common structure of all Party elements

There has to be at least one name and one address element inside a party element, but to offer a wider range of descriptive information the number of such elements is not limited. That means that you can add nicknames, names at birth etc. as well as business and other addresses. The nature of names and addresses can be indicated by optional attributes, the admissible values are

for names:

```

<xsd:enumeration value="SMFAliasOrOther"/>
<xsd:enumeration value="indiv"/>
<xsd:enumeration value="alias"/>
<xsd:enumeration value="nick"/>
<xsd:enumeration value="aka"/>
<xsd:enumeration value="dba"/>
<xsd:enumeration value="legal"/>
<xsd:enumeration value="atbirth"/>
  
```

and for addresses:

```

<xsd:enumeration value="residentialOrBusiness"/>
<xsd:enumeration value="residential"/>
<xsd:enumeration value="business"/>
<xsd:enumeration value="registeredOffice"/>
  
```

Most of this will be self-explanatory, let us just mention that aka stands for “also known as”, dba for “doing business as”. SMFAliasOrOther is an attribute value that should only be used if the document is generated by a bridging program from a SMF record. If there is an entry in the field group “alias or other” in the SMF record (a group within the beneficial owner group which holds all of “aka”, “dba” etc.), the bridging program will not be able to decide which kind of name that is and therefore will translate just into “SMFAliasOrOther”.

We will go into the detailed structure of names and addresses later.

The residence country (in the relevant time period), to be represented in element ResCountryCode by its ISO 3166 two-byte alpha code, is considered to be a property of the party, not an address, although it is most likely that it will coincide with at least one of the address country codes for this party. To be sufficiently general, the element had to

be left optional, even if information about someone with unknown residence country will probably less than helpful.

Another important item of the party description is formal identifiers (to be entered in elements PartyId), for which 3 optional entries are provided. The idea is to give whatever official identification “numbers” are known by the sending country. PartyId elements are declared as shown in Schema Fragment 5:

```
<xsd:complexType name="PartyId_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="partyIdType" type="partyIdType_Type" use="required"/>
      <xsd:attribute name="issuedBy" type="xsd:string" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

SchemaFragment 5

The attribute partyIdType is to distinguish between the kinds of identifiers like Tax Identification Number (TIN), Tax File Number (TFN) and others. To-date TIN, TFN and IdNo are defined as valid entries. It is required to add another attribute (issuedBy) to describe the body that has issued the identifier to the party. In the case of a TIN this should be just the country code of the issuing country, in other cases a non-formalised entry will be adequate.

To even better describe and hopefully identify the party, an optional element PersData can take more information, depending on the type of the party (individual or legal). The content will become clear from Figure 8:

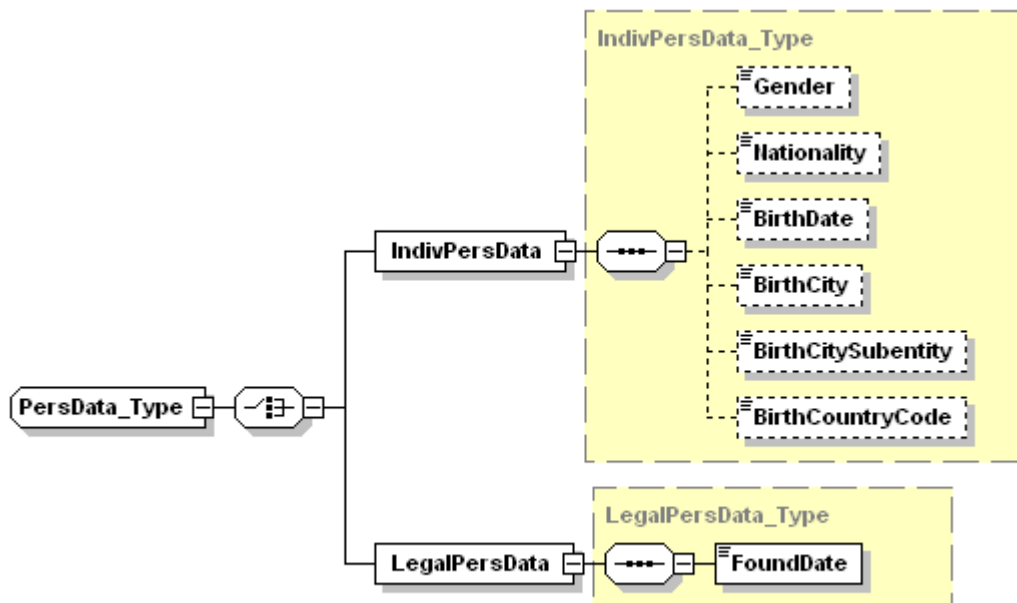


Figure 8: Additional identifying data about a party

In the following paragraphs we will now see how names and addresses are dealt with inside the party structure of STF.

5.6 Names

Here is the broad picture:

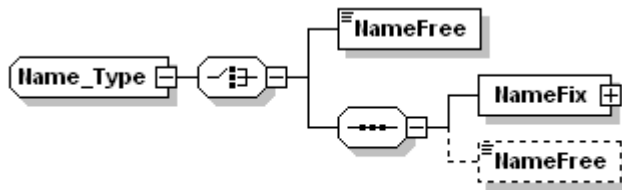


Figure 9: Name structure

It will be noticed that a name can be either a NameFree element, or a NameFix element, or a sequence of both. NameFree will be used to deal with the common situation that it is not really clear for the sending country what are the roles of different particles in a sequence of words that constitute the name of a party. In such cases it may be better to leave it to the receiving country to sort that out, as it may be better acquainted with the name structure of the party. Ideally of course the name of the party is well structured into parts that are identified by the sending country. To serve cases where a structured name (NameFix) can be given, but only with some doubt as to the validity of the break-down into its parts, the sending side may choose to provide a NameFree in addition.

Widely accepted international standards for name structure are only just emerging and for individuals STF has chosen to adhere – although not too closely – to the CIQ standard for names (CIQ: Customer Information Quality, an OASIS family of standards), resulting in the following structure:

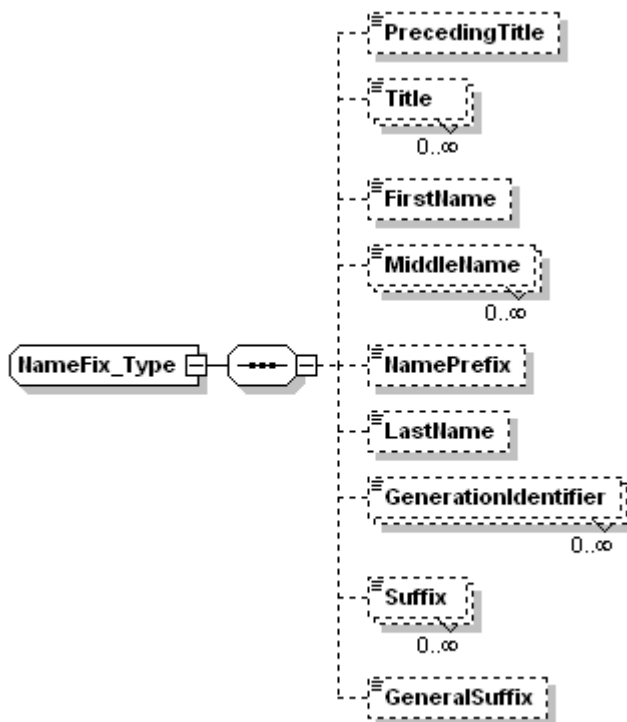


Figure 9: Structured Name in STF

All elements in this structure are optional, as there is no guarantee that a particular one will definitely be present in all cases. Of course there will have to be at least one entry for the name to be useful. Following CIQ, FirstName, MiddleName, NamePrefix, and LastName designate exactly what their names say, that is: the sequence in “normal” usage. The meaning e.g. of the first part of a name may, however, vary from one cultural environment to another. Therefore all of the elements mentioned have to be qualified by an attribute, which is called xnlNameType (xNL is the standard for names in the CIQ family of standards). For the time being there is no predefined set of values for xnlNameType, as also CIQ leaves this to the user. “given Name”, “family name” etc. may be values you may want to use. For legal entities always the free form shall be used for the name; there does not seem to be any useful standardised way of breaking down such names into well-defined parts.

5.7 Addresses

The top level view on addresses is this:

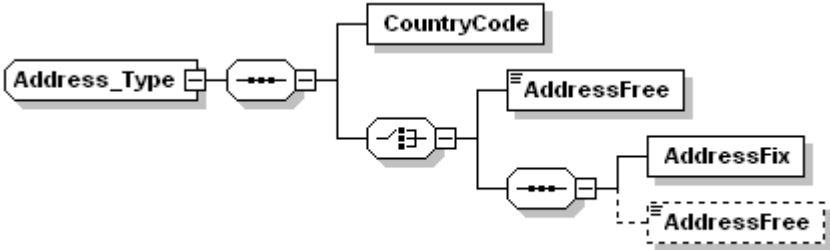


Figure 10: Address structure

Like for names the address can be either an AddressFree element, or an AddressFix element, or a sequence of both. The country code is left outside these structures, as it is be a well-discernable field that never should be imbedded (and hidden) in an unformatted character sequence like in AddressFree.

For addresses more or less the same remarks apply as for names: AddressFree will be used to deal with the common situation that it is not really clear for the sending country what are the roles of different particles in a sequence of words that constitute the address. Also in such cases it may be better to leave it to the receiving country to sort that out, as it may be better acquainted with the address structure of the party. Ideally of course the address of the party is well structured into parts that are identified by the sending country. To serve cases where a structured address (AddressFix) can be given, but only with some doubt as to the validity of the break-down into its parts, the sending side may choose to provide a AddressFree in addition.

Widely accepted international standards for address structure are only just emerging. For STF it has been considered to mimic the CIQ standard for addresses as it did with names. It was found, however, that xAL, the address standard inside CIQ, has gained its extreme flexibility and wide applicability by a degree of complexity that did not seem adequate for STF. This design decision was flagged as “to be monitored”, as possible widespread use of xAL in OECD member countries may well suggest reconsidering the design. For the time being, addresses in their fixed format are structured like this in STF:

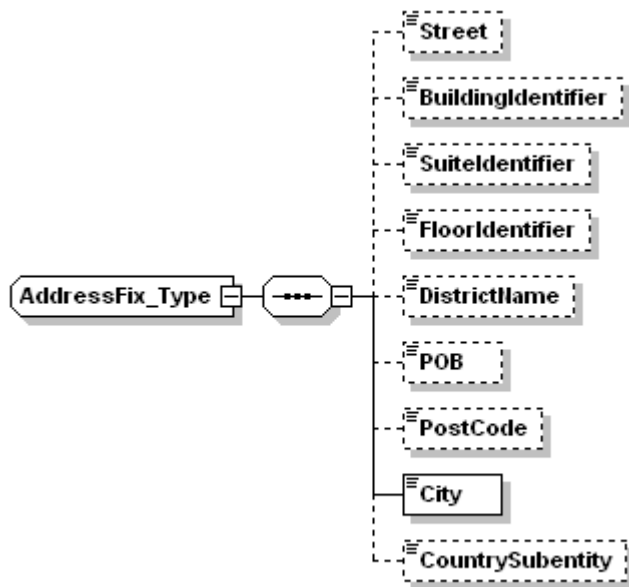


Figure 11: Structured Address in STF

The only mandatory element in this structure is the name of the city. Other address parts shall be given as available.

6. Where is detailed advice for all of the elements and their content to be found?

The central source for all guidance concerning STF is the set of STF XML schemas. Annotations are to be found in the schemas for more or less all of the relevant elements and data types. They are in many cases just replica of the comments to SMF fields in the SMF Manual. As an XML schema is not readily readable by non-IT people and as even for those it may be cumbersome to find a specific piece of documentation in a lengthy schema, comments have been extracted by an automated procedure from the schemas and an “Electronic Manual” has been generated in HTML-format. Users are thus able to find guidance simply by directing their browser to the relevant URL. (---- or a mirror in a country’s own web site.)

The Electronic Manual presents the user two columns of information. The left side column is an indented list of the elements and attributes that constitute the most comprehensive STF_OECD document in accordance with the schema. The right hand side contains the annotations to all data types defined. There are two kinds of interactivity provided: On “mouse-over” at an element in the left hand side hierarchy comments concerning the element are displayed. On “click” over left hand side elements the right hand side is positioned to display comments concerning the data type of the element. (Not all elements possess mouse-over comments nor do all data types possess clickable comments.)

The (left column) structural image of an STF document in the Electronic Manual cannot totally repeat all of the structure information of the schemas. For instance it does not contain an indication whether an element is optional or mandatory. If the complete picture is needed, the reference should be either the schema itself (Appendix 1) or the comprehensive diagram in Appendix 2.

For users with an IT background who want a really comprehensive documentation there is also an automatically generated HTML based documentation which takes in account everything from the schemas but is better suited for reading than the schema code itself. A Word version is attached as Appendix 3.

To avoid even more pages to be generated the country code and currency code lists have been shortened in the printed versions to contain just a few examples.

7. How is coexistence between SMF and STF guaranteed?

Even if XML is the worldwide acknowledged standard for transmission of data between systems and many if not most of OECD member states have an e-Government policy including XML as a preferred standard, countries with a working SMF environment may be reluctant to spend resources for a migration to STF. On the other hand, countries that want to introduce automatic means for international information exchange in taxation just now will probably not want to introduce methods that reflect the IT situation of the 1990s. To adequately deal with that transient situation, bridging programs have been written that transform SMF records into STF_DIRECT documents and vice-versa.

These programs are XSL transformations and have the self-explanatory names

smf2stf
stf2smf

It will certainly be understood that neither OECD nor the authors of the bridging programs can be held liable for any errors of these programs or consequences of such errors. The programs are offered as a support to smooth migration, responsibility for the use of the programs and the data being exchanged stays with the users. Therefore this DISCLAIMER is included in the transformation code:

THIS TRANSFORMATION HAS BEEN WRITTEN AND TESTED WITH CARE. THERE WILL BE, HOWEVER, NO GUARANTEE WHATSOEVER REGARDING ITS CORRECTNESS. ANYONE USING THIS TRANSFORMATION WILL DO THIS UNDER HIS OR HER OWN RESPONSIBILITY AND BEFORE USING IT WILL HAVE TO TEST IT AS CONSIDERED NECESSARY. NO LIABILITY WILL BE ACCEPTED BY OECD, THE OECD TIES GROUP, OR THE AUTHORS OF THIS TRANSFORMATION FOR ANY DIRECT OR INDIRECT DAMAGE THAT MAY RESULT FROM USING THIS TRANSFORMATION. THIS TRANSFORMATION MAY BE USED AND CHANGED FREELY IF AND ONLY IF THESE CONDITIONS ARE ACCEPTED.

It has to be remarked that - mostly due to the slightly enhanced generality of STF compared to SMF - bridging can be done nearly 100%, but not exactly 100%. In the following paragraphs we will explain the issues to be noticed.

I. Bridge from SMF to STF

7.0 Bridging can be done either at the sending or at the receiving side of the SMF file. There is some merit and demerit to either choice. Bridging at the source will enable the sending country to validate the resulting STF file against the schemas and thus filter out any irregularities early in the process. The sender will be better prepared to deal with the file that they have written themselves and some problems concerning readability may be avoided when an STF file in UTF-8 encoding is transmitted. But it also means that the sending country has to keep record of countries that use the STF. As long as the STF has not become the prevailing format it seems best to leave it up to the parties involved in the exchange to decide who will do the bridging.

7.1 Bridging is done via an XSL Transformation. As such transformations operate on XML files, a preparatory task is to format the SMF file into an XML file. The transformation program expects the input wrapped by root element tags "SMFFile", "/SMFFile" and the

records made into “Record” elements, i.e. every record has to be surrounded by “Record”, “/Record” tags. Also depending on the XSL transformation procedure a processing instruction
`<?xml-stylesheet type="text/xsl" href="pppp\smf2stf01.2.xsl"?>`
 may have to be added at the beginning, with pppp to be replaced by the path to the transformation file. It may also be the case that code transformation from mainframe encoding – preferably to UTF-8 – has to be done prior to the XSL. These preparations will be rather straightforward but have to be done according to the individual situation at the member state’s site and the SMF file at hand and are therefore not supported by the OECD bridging system.

7.2 SMF files do not have an equivalent to the STF message header “MessageSpec”. Therefore there is no source in such files from which the MessageSpec element could be automatically generated from. It is therefore the duty of the person preparing the bridging program’s execution to enter the relevant data into the XSL code before the transformation is done. Here is the part of the XSL that has to be adapted:

```

<!-- ***** -->
<!-- ***** To be edited before Transformation ***** -->
  <MessageSpec>
    <SendingCountry>Country Code</SendingCountry>
    <ReceivingCountry>Country Code</ReceivingCountry>
    <Warning>Legal Information</Warning>
    <Contact>Who to contact for this message</Contact>
    <MessageRefId></MessageRefId> <!-- recommendation: leave void -->
    <TaxYearList>list of tax year ends in form: 2002-12-31</TaxYearList>
  </MessageSpec>
<!-- ***** -->
<!-- ***** -->

```

7.3 In SMF, referencing records that were sent before (for correction or repetition cases) is done on the record level only, there is no message (file) identifier in SMF. Therefore for STF files generated from SMF, matching between records cannot be based on the combination of message identifier and document identifier. It is therefore recommended not to attribute message identifiers to STF messages generated from SMF files. (The empty element MessageRefId has to stay there in order to make the document valid.) All matching should thus remain unaffected, though it will not be possible to refer to the message itself by an unambiguous identifier.

7.4 If for the beneficial owner in the SMF record something is entered in the field group “Alias Or Other”, a “Name” child element is generated for the STF “RecipientBeneficialOwner” element with the value “SMFAliasOrOther” for the “nameType” attribute.

7.5 Any entries in the SMF field group “In Care Of Person” will be lost. (It seems that nobody has ever made use of this feature in SMF.)

7.6 If in the SMF record there are erroneously for name or address type other entries than ‘0’ (for fixed format) and ‘1’ (for free format) the bridging program will assume the content of the following field(s) to be in free format and transform accordingly.

7.7 If in SMF a gender of “M” (male) or “F” (female) is given for the beneficial owner, there will be a child element “PersData” for the beneficial owner element in STF with that information and – if present in SMF – information about the city of birth. The entry “N” (non individual) will result in *no* PersData element, as there had to be mandatory content, which is, however, not available in SMF.

II. Bridge from STF to SMF

7.8 An STF document can supply multiple party identifiers for all parties, they can be TINs, but can also other kinds of identifiers. SMF is only supposed to have TINs as identifiers for parties, so any other identifiers in an STF document will be lost by the bridging transformation. For all parties SMF has two TIN fields along with the respective fields for country codes designating the issuing state of the TINs. There is a special situation for the beneficial owner party, as here SMF explicitly asks for the first TIN (and country code) to belong to the residence country and the second to the source country. The bridging program does the following:

- if the element RecipientBeneficialOwner contains a ResCountryCode element, the first TIN field of the bridging result will only contain an entry if there is a TIN PartyId element for the beneficial owner with this country code as the issuedBy attribute. The second TIN field of the result record will contain the data from another TIN PartyId element for the beneficial owner (if any), but there is no test executed whether this will belong to the source country;
- if on the other hand the element RecipientBeneficialOwner does not contain a ResCountryCode element (which is optional in STF), the TIN fields of the result will just contain the data from any two TIN PartyId elements (as far as existent in the STF document).

7.9 An STF document can contain multiple names for all parties. SMF can have two names (including “alias or other”) for the recipient beneficial owner party and only one name for the other parties. In bridging, the main name field for the beneficial owner party is filled with the first STF name found which has no nameType attribute or where the nameType attribute is “legal” or “individual”; it is left blank if no such element exists (that is, all Name elements present are nameType-d as “aka”, “dba” etc.). Name elements exceeding the number of two (for the beneficial owner) or one (for all other parties) are lost by bridging, with the one exception of a Name element with nameType attribute “at birth”: atbirth-names are appended to the name inside the main name field with the addition of “at birth”. Also in the case that both a free form and a fixed form name are given one (the fixed form) is lost.

7.10 An STF document can contain PaymentDate elements within every Payment element. There is only one field for a payment date in SMF. This field will be filled from the Payment element with ‘gip’ (gross income paid) as value for the paymentQlf attribute, if such element exists and has a PaymentDate (which is optional for all payments). If this does not result in filling the field, the next source to look for the payment date will be the ‘nip’ (net income paid) Payment element. If neither gip nor nip has dates, the date field in the SMF record will be left blank.

8. Examples of Elements and Messages

8.1 Examples of the simplest and the most complete MessageSpec element

```
<MessageSpec>  
  <Warning>Only to be used in conformance with our Agreement</Warning>  
  <Contact>Rosalie Sender mailto: Rosalie@sender.gov.de</Contact>  
  <MessageRefId>123123</MessageRefId>  
  <TaxYearList>2004-12-31</TaxYearList>  
</MessageSpec>
```

```

<MessageSpec>
  <SendingCountry>GB</SendingCountry>
  <ReceivingCountry>US</ReceivingCountry>
  <Warning>Please do not use this</Warning>
  <Contact>Mark Anthony phone 007 135 791</Contact>
  <MessageRefId>1234567</MessageRefId>
  <TaxYearList>1999-04-05 2002-04-05</TaxYearList>
</MessageSpec>

```

8.2 Examples of a DocSpec element heading a “new” document and of one correcting another that was sent before (document 1000001 in the message belonging to the first MessageSpec above)

```

<DocSpec>
  <DocTypeIndic>1</DocTypeIndic>
  <DocRefId>987654</DocRefId>
</DocSpec>

```

```

<DocSpec>
  <DocTypeIndic>2</DocTypeIndic>
  <DocRefId>5656565</DocRefId>
  <CorrMessageRefId>123123</CorrMessageRefId>
  <CorrDocRefId>1000001</CorrDocRefId>
</DocSpec>

```

8.3 Examples of Name elements

Name element in free format belonging to an individual person party

```

<Name nameType="indiv">
  <NameFree>Arndt Liesen</NameFree>
</Name>

```

Name element in free format belonging to a legal entity

```

<Name nameType="legal">
  <NameFree>Arndt Liesen IT consultancy and training Incorporated</NameFree>
</Name>

```

Name element in fixed format belonging to an individual

```

<Name nameType="indiv">
  <NameFix>
    <PrecedingTitle>Her Excellency</PrecedingTitle>
    <Title>Ms</Title>
    <FirstName xnlNameType="Given Name">Mary</FirstName>
    <MiddleName xnlNameType="Middle Initial">R</MiddleName>
    <NamePrefix xnlNameType="Prefix">de</NamePrefix>
    <LastName xnlNameType="Family Name">Smith</LastName>
    <GenerationIdentifier>II</GenerationIdentifier>
    <Suffix>PhD</Suffix>
    <GeneralSuffix>Retired</GeneralSuffix>
  </NameFix>
</Name>

```

8.4 Examples of Address elements

Address element of a business site in Germany in free format

```
<Address legalAddressType="business">
  <CountryCode>DE</CountryCode>
  <AddressFree>Friedhofstrasse 1 53225 Bonn</AddressFree>
</Address>
```

Same address in fixed format

```
<Address legalAddressType="business">
  <CountryCode>DE</CountryCode>
  <AddressFix>
    <Street>Friedhofstrasse 1</Street>
    <PostCode>53619</PostCode>
    <City>Bonn</City>
  </AddressFix>
</Address>
```

Complex residential address in fixed format (example adapted from an OASIS CIQ standard example, describing the Australian address

block 2, RIPPON BUILDING Level 12, Suite 1A
(47 Kingston Avenue North, North Ryde, NSW 2113, Australia)

```
<Address legalAddressType="residential">
  <CountryCode>AU</CountryCode>
  <AddressFix>
    <Street> 47 Kingston Avenue North</Street>
    <BuildingIdentifier>Block 2, Rippon Building</BuildingIdentifier>
    <SuiteIdentifier>Suite 1A</SuiteIdentifier>
    <FloorIdentifier>Level 12</FloorIdentifier>
    <City>North Ryde</City>
    <CountrySubentity>NSW</CountrySubentity>
  </AddressFix>
</Address>
```

8.5 Examples of elements of Party type

A beneficial owner Party element representing an individual person

```
<RecipientBeneficialOwner oecdLegalType="01">
  <ResCountryCode>DE</ResCountryCode>
  <PartyId partyIdType="TIN" issuedBy="DE">777666555</PartyId>
  <Name nameType="indiv">
    <NameFree>Arndt Liesen</NameFree>
  </Name>
  <Address legalAddressType="residential">
    <CountryCode>DE</CountryCode>
    <AddressFix>
      <Street>Mystreet 77</Street>
      <PostCode>77777</PostCode>
      <City>Mycity</City>
    </AddressFix>
  </Address>
  <PersData>
    <IndivPersData>
      <Gender>M</Gender>
      <Nationality>DE</Nationality>
      <BirthDate>1939-04-16</BirthDate>
      <BirthCity>Duisburg - Germany</BirthCity>
    </IndivPersData>
  </PersData>
</RecipientBeneficialOwner>
```

A payer Party element (address example adapted from an OASIS CIQ standard example)

```
<ActualPayer oecdLegalType="02">
  <ResCountryCode>JP</ResCountryCode>
  <PartyId partyIdType="TFN" issuedBy="JP Tax Authority">1234567</PartyId>
  <Name nameType="legal">
    <NameFree>The very big Payer of Income from Interest Inc.</NameFree>
  </Name>
  <Address legalAddressType="residentialOrBusiness">
    <CountryCode>JP</CountryCode>
    <AddressFree>Japan 530-0001 Osaka Prefecture Oasaka City North Ku Plum Rice Field
    1-2-2 the 2nd Building before the Oasaka Station </AddressFree>
  </Address>
</ActualPayer>
```

8.6 Examples of PaymentData elements

A gross interest payment (OECD payment type 11) of EUR 2000 was effected in tax year 2003

```
<PaymentData>
  <TaxYearEnd>2003-12-31</TaxYearEnd>
  <OECDPaymentType>11</OECDPaymentType>
  <Payment paymentQlf="gip">
    <MonAmnt currCode="EUR">2000</MonAmnt>
  </Payment>
</PaymentData>
```

In the tax year ending on April 5 2001 these payments were effected, qualified as “Director’s Fees” (OECD payment type 16), but more precisely qualified by a country specific payment type of “P47a” according to some classification scheme called “UK2001”:

- gross payment of 4000 pounds
- reduced to net payment of 2000 pounds
- followed by a tax refund of 1000 pounds.

```
<PaymentData>
  <TaxYearEnd>2001-04-05</TaxYearEnd>
  <OECDPaymentType>16</OECDPaymentType>
  <SpecificPaymentType specificPaymentTypeQlf="UK2001">P47a</SpecificPaymentType>
  <Payment paymentQlf="gip">
    <PaymentDate>2000-06-31</PaymentDate>
    <MonAmnt currCode="GBP">4000</MonAmnt>
  </Payment>
  <Payment paymentQlf="nip">
    <PaymentDate>2000-06-31</PaymentDate>
    <MonAmnt currCode="GBP">2000</MonAmnt>
  </Payment>
  <Payment paymentQlf="trf">
    <PaymentDate>2000-09-10</PaymentDate>
    <MonAmnt currCode="GBP">1000</MonAmnt>
  </Payment>
</PaymentData>
```

For this the schema file for country specific payment codes, countryspecificitytypes_v1.xsd, would have had to be edited like this:

```
...
<xsd:complexType name="SpecificPaymentType_Type">
  <xsd:annotation>
    <xsd:documentation>Type for explanation of a payment by a code that is specific for the UK (version of 2001)
    special codes used are:
```

- **S20 interest from extremely large deposits**
- ...
- **P47a fees for directors of chains of nightclubs**
-

```

</xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="SpecifPaymentType_Type ">
      <xsd:attribute name="SpecifPaymentTypeQif" type="xsd:string" fixed="UK2001"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name=" SpecifPaymentType_Type ">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="S20"/>
    ....
    <xsd:enumeration value="P47a"/>
    ....
  </xsd:restriction>
</xsd:simpleType>

```

8.7 A complete message containing two documents for different tax years, one a correction to another document assumed to be sent before

```

<STF_OECD xmlns="urn:oe.cd:ties:stf:v1" version="1.0">
  <MessageSpec>
    <SendingCountry>US</SendingCountry>
    <ReceivingCountry>DE</ReceivingCountry>
    <Warning>Only to be used in conformance with our Agreement</Warning>
    <Contact>Rosalie Sender mailto: Rosalie@sender.gov.us</Contact>
    <MessageRefId>US20023-4</MessageRefId>
    <TaxYearList>2003-12-31 2002-12-31</TaxYearList>
  </MessageSpec>
  <STF_DIRECT version="1.0">
    <DocSpec>
      <DocTypeIndic>1</DocTypeIndic>
      <DocRefId>987654</DocRefId>
    </DocSpec>
    <RecipientBeneficialOwner oecdLegalType="01">
      <ResCountryCode>DE</ResCountryCode>
      <PartyId partyIdType="TFN" issuedBy="DE">32/001/47133</PartyId>
      <PartyId partyIdType="TIN" issuedBy="US">123456433</PartyId>
      <Name nameType="indiv">
        <NameFix>
          <PrecedingTitle>Her Excellency</PrecedingTitle>
          <Title>Ms</Title>
          <FirstName xnlNameType="Given Name">Mary</FirstName>
          <MiddleName xnlNameType="Middle Initial">R</MiddleName>
          <NamePrefix xnlNameType="Prefix">de</NamePrefix>
          <LastName xnlNameType="Family Name">Smith</LastName>
          <GenerationIdentifier>II</GenerationIdentifier>
          <Suffix>PhD</Suffix>
          <GeneralSuffix>Retired</GeneralSuffix>
        </NameFix>
      </Name>
      <Name nameType="aka">
        <NameFree>Mary the Belle</NameFree>
      </Name>
      <Name nameType="atbirth">
        <NameFree>Marie Dupont</NameFree>
      </Name>
      <Address legalAddressType="residential">
        <CountryCode>DE</CountryCode>
        <AddressFix>
          <Street>Friedhofstrasse 1</Street>
          <PostCode>53225</PostCode>
        </AddressFix>
      </Address>
    </RecipientBeneficialOwner>
  </STF_DIRECT>
</STF_OECD>

```

```
        <City>Bonn</City>
      </AddressFix>
    </Address>
  <PersData>
    <IndivPersData>
      <Gender>F</Gender>
      <Nationality>FR</Nationality>
      <BirthDate>1937-08-13</BirthDate>
      <BirthCity>Paris</BirthCity>
      <BirthCitySubentity>Montmartre</BirthCitySubentity>
      <BirthCountryCode>FR</BirthCountryCode>
    </IndivPersData>
  </PersData>
</RecipientBeneficialOwner>
<RecipientAgentOrIntermediary oecdLegalType="01">
  <ResCountryCode>DE</ResCountryCode>
  <Name nameType="legal">
    <NameFree>The Mary the Belle Trust</NameFree>
  </Name>
  <Address legalAddressType="business">
    <CountryCode>DE</CountryCode>
    <AddressFree>53221 Bonn</AddressFree>
  </Address>
</RecipientAgentOrIntermediary>
<ActualPayer oecdLegalType="02">
  <ResCountryCode>US</ResCountryCode>
  <PartyId partyIdType="TIN" issuedBy="US">99999999</PartyId>
  <Name nameType="legal">
    <NameFree>Grey Dancers Great Performances</NameFree>
  </Name>
  <Address legalAddressType="business">
    <CountryCode>US</CountryCode>
    <AddressFix>
      <Street>100 Broadway</Street>
      <City>NewYork</City>
      <CountrySubentity>NY</CountrySubentity>
    </AddressFix>
  </Address>
</ActualPayer>
<PaymentData>
  <TaxYearEnd>2003-12-31</TaxYearEnd>
  <OECDPaymentType >17</OECDPaymentType>
  <Payment paymentQlf="gip">
    <PaymentDate>2003-07-06</PaymentDate>
    <MonAmnt currCode="USD">7100</MonAmnt>
  </Payment>
</PaymentData>
<OtherInfo>Please report back on matching with a real person</OtherInfo>
</STF_DIRECT>
<STF_DIRECT version="1.0">
  <DocSpec>
    <DocTypeIndic>2</DocTypeIndic>
    <DocRefId>564534</DocRefId>
    <CorrMessageRefId>US10001-7</CorrMessageRefId>
    <CorrDocRefId>561212</CorrDocRefId>
  </DocSpec>
  <RecipientBeneficialOwner oecdLegalType="03">
    <ResCountryCode>DE</ResCountryCode>
    <Name nameType="legal">
      <NameFree>The Big Earners Partnership</NameFree>
    </Name>
    <Address legalAddressType="residentialOrBusiness">
      <CountryCode>DE</CountryCode>
      <AddressFree>Somewhere in Frankkfurt, Germany</AddressFree>
    </Address>
  </RecipientBeneficialOwner>
  <PayerAgentOrIntermediary oecdLegalType="04">
    <ResCountryCode>US</ResCountryCode>
    <PartyId partyIdType="TIN" issuedBy="US">124534</PartyId>
    <Name nameType="legal">
      <NameFree>First Banking for Nothing</NameFree>
    </Name>
    <Address legalAddressType="unspecified">
```

```

    <CountryCode>US</CountryCode>
    <AddressFree>77 Gold Avenue, Las Vegas, Nevada</AddressFree>
  </Address>
  <PersData>
    <LegalPersData>
      <FoundDate>2002-01-01</FoundDate>
    </LegalPersData>
  </PersData>
</PayerAgentOrIntermediary>
<PaymentData>
  <TaxYearEnd>2002-12-31</TaxYearEnd>
  <OECDPaymentType>11</OECDPaymentType>
  <SpecificPaymentType specificPaymentTypeQlf="US special">11-11</SpecificPaymentType>
  <Payment paymentQlf="gip">
    <PaymentDate>2002-01-02</PaymentDate>
    <MonAmnt currCode="EUR">900000001</MonAmnt>
    <TaxRate>30.5</TaxRate>
  </Payment>
  <Payment paymentQlf="trf">
    <PaymentDate>2003-03-15</PaymentDate>
    <MonAmnt currCode="USD">100000000</MonAmnt>
  </Payment>
</PaymentData>
  <OtherInfo>US-special income type 11-11 is interest from doubtful source</OtherInfo>
</STF_DIRECT>
</STF_OECD>

```

9. What artefacts are available for STF?

All of the following is available in electronic form at (*URL to be inserted*).

9.1 This introductory manual and its appendices

STFexplained.doc
 STFexpl-app1-schemas.doc
 STFexpl-app2-diagram.png
 STFexpl-app3-tecdocu.doc

9.2 The STF schema files

stfdirect-1.0.xsd
 stftypes-1.0.xsd
 isotypes_v1.xsd
 oecdtypes_v1.xsd
 specifictypes_v1.xsd

9.3 The bridging programs (XSL stylesheets)

stf2smf-1.0.xsl
 smf2stf-1.0.xsl

9.4 The electronic manual in html (hierarchic structure with links to annotations)

STFmanual.html (the main document presenting three frames)
 STFhead.html
 STFstru.html
 STFdoc.html

9.5 The complete generated documentation for the schema-system in html

stf-1.0-generatedDocu.html

stf-1.0-generatedDocu_pxx.png (93 png-files for use by stf-1.0-generatedDocu.html)

