OECD STANDARDISED FORMAT FOR AUTOMATIC EXCHANGE OF INFORMATION FOR TAX PURPOSES

Background

1. Article 26 of the OECD Model Tax Convention allows for tax information to be exchanged by competent authorities in three ways: on-request, automatically and spontaneously. Information suitable for automatic exchange is typically bulk information comprising many individual cases of the same type, usually consisting of details of income arising from sources in the supplying state where such information is available periodically under that state's own system and can be transmitted automatically on a routine basis. Automatic exchange of information requires standardisation of formats in order to be efficient.

2. The OECD's first step towards standardising the presentation of this type of information resulted in the design, in 1981, of a paper-based form for use in automatic (routine) exchange. The subsequent proliferation of electronic data processing capabilities within tax administrations led to the development of the OECD Standard Magnetic Format (SMF) for automatic exchange of information in 1992, followed by a revised version in 1997[1] (the version currently in use). The SMF contains similar information to the paper-based form but is designed for transmission using electronic media. This enhances data matching in the receiving country and facilitates the exchange of information concerning a large number of taxpayers in a single transmission. The SMF record layout includes fields allocated to the:

- recipient beneficial owner, his agent or intermediary;
- payer of the income, his agent or intermediary;
- residence and source country TINs;
- income derived (tax year, date paid, type of payment, currency, gross and net amount paid, tax withheld, amount refunded etc.).

The Taxation Information Exchange Systems (TIES) Sub-Group has been monitoring the implementation of the SMF since its inception.[2]

Issues

3. The SMF was originally designed for the transmission of information on magnetic tapes. However, magnetic tapes are no longer universally used and for some years countries have been using diskettes and CD-ROMs without formal background to transmit taxpayer information automatically. Moreover the conventional record layout is somewhat inflexible and not a suitable base for future modifications and extensions.

4. In this regard, the TIES Sub-Group has developed a "new generation" transmission format for automatic exchange to replace the SMF. The new format is called the Standard Transmission Format (STF)

---

[1] Refer to Council Recommendations C(92)50/FINAL (23 July 1992) and C(97)30/FINAL (10 July 1997).

[2] Refer to the 2002 Survey on the Implementation and use of the OECD SMF (DAFFE/CFA/WP8/TIES(2002)5/CONF) for further information.

and is based on extensible markup language (XML[3]), a document markup language widely used in today's information technology for its many advantages, e.g.:

- separation of the content of a message from any display structure;
- readability both by humans and machines;
- modularity and flexibility;
- ability to check the conformance of documents with the "contract" about its structure;
- availability of a host of tools.

Annex 2 sets out the schemas of the STF (this information is primarily for the benefit of staff involved with the information technology aspects of exchange of information).

5. The STF is intended to co-exist with the SMF rather than to replace it in the near future. The SMF remains an effective tool and the TIES Sub-Group will continue to support its use for the foreseeable future. Migrating to the STF in the short term is not a practical option for most countries that currently use the SMF. However, the majority of OECD countries currently do not use the SMF and the adoption of the STF would allow those countries to move straight to the best available technology when developing the capacity to exchange information automatically. In this regard bridging programmes have been developed to achieve conversion between the two formats, thus enabling treaty partners to engage in bilateral automatic exchange notwithstanding that they might each use a different standard format. Features of the STF that exceed the capabilities of the SMF will not be bridgeable. However, there are currently only a few such features (of minor importance) because compatibility with the SMF was a key design goal for the first version of the STF. Thus the impact for countries that currently use the SMF will be minimal.

6. The STF is ultimately intended to form part (Level 1: Transmission Media) of a framework known as *OECD Standards on Exchange of Information in Taxation* (SEIT). The TIES Sub-Group is presently developing this framework (see Annex I) in relation to three main practical aspects of exchange of information:

- Transmission (how information is physically sent and received by competent authorities: refer to CTPA/CFA/WP8/TIES(2004)1/CONF for a discussion of the electronic transmission of encrypted taxpayer information)
- Security (the protection of confidentiality)
- Format (how information is presented)

Proposal

7. The Committee on Fiscal Affairs agreed to Working Party No. 8's proposal that the STF be accepted as the new OECD standard format for automatic exchange of information, subject to the following conditions:

- Countries that currently use the SMF can migrate to the STF but for the foreseeable future will not be required to migrate to the STF;
- The TIES Sub-Group will continue to monitor and support the use of the SMF for the foreseeable future;
- The bridging programmes will enable the two formats to operate in parallel.

---

[3] XML: a technical language for describing documents containing structured information. The term "extensible" refers to a system that can be enlarged by addition rather than by complete replacement.

ANNEX 1

OECD STANDARD FOR EXCHANGE OF INFORMATION IN TAXATION (SEIT)				
	On-request and spontaneous exchange		**Automatic exchange**	
	Physical exchange	**Electronic exchange**	**Physical exchange**	**Electronic exchange**
Level 1 **Transmission media**	Paper documents transmitted via normal mail, commercial courier, diplomatic bag etc.	Email attachment	Magnetic tape, diskette, CD Rom (or DVD[4]) transmitted via normal mail, commercial courier, diplomatic bag etc.	Email attachment[5] Electronic file transfer[6] SOAP[7]
Level 2 **Encryption**	No encryption of paper documents.	Standard Transmission Encryption[8] [see DAFFE/CFA/ WP8/TIES(2003)5/CONF for encryption and key management procedures]	Standard Transmission Encryption (see footnote 13) [see DAFFE/CFA/ WP8/TIES(2003)5/CONF for encryption and key management procedures]	Standard Transmission Encryption (see footnote 13) [see DAFFE/CFA/ WP8/TIES(2003)5/CONF for encryption and key management procedures]
Level 3 **Content Format**	N/A for paper documents.	PDF, JPG or TIFF files for scanned documents. RTF for electronically stored documents.[9]	Standard Magnetic Format (SMF); or Standard Transmission Format (STF)	Standard Magnetic Format (SMF); or Standard Transmission Format (STF)

4 DVDs are not generally used by competent authorities but can hold much more data than CD-ROMs.

5 Automatic exchange via email is possible (subject to file size) but is generally not considered necessary.

6 A potential future method of transmission.

7 A potential future method of transmission. SOAP (Simple Object Access Protocol) is a communications protocol that enables objects created under different systems to invoke each other's methods by exchanging plain text messages in an XML based format over an ordinary HTTP connection. For exchange of information purposes, SOAP can also serve just for the exchange of plain text messages (preferably in an XML format).

8 Gnu Privacy Guard was the encryption software used in the TIES Sub-Group pilot on electronic exchange of Category 3 information. It uses the Pretty Good Privacy (PGP) standard and is compatible with commercial PGP products.

9 PDF (Portable Document Format); JPG (Joint Photographic Group standard format); TIFF (Tagged Image File Format); RTF (Rich Text Format).

ANNEX 2: STF SCHEMAS

MAIN SCHEMA STFDIRECT


```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="urn:oecd:ties:stf:v1" xmlns="urn:oecd:ties:stf:v1" xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xsd:include schemaLocation="stftypes-1.0.xsd"/>
    <xsd:complexType name="STF_Direct_Type">
        <xsd:sequence>
            <xsd:element name="DocSpec" type="DocSpec_Type">
             <xsd:annotation>
                <xsd:documentation>General information concerning this document - document specification</xsd:documentation>
             </xsd:annotation>
            </xsd:element>
            <xsd:element name="RecipientBeneficialOwner" type="Party_Type">
             <xsd:annotation>
                <xsd:documentation>This is the Recipient Beneficial Owner party of stfdirect. The recipient beneficial owner is the person (legal person or individual), resident of a contracting State, that is entitled to the income for tax purposes and has the benefit thereof, taking into account the economic, legal, factual, and other relevant circumstances (e.g. the relevant double taxation treaty) under which the income is received.
</xsd:documentation>
             </xsd:annotation>
            </xsd:element>
            <xsd:element name="RecipientAgentOrIntermediary" type="Party_Type" minOccurs="0">
             <xsd:annotation>
                <xsd:documentation>Recipient Agent or Intermediary party of stfdirect. This is a party that received the payment in question but is known not to be the recipient beneficial owner (e.g., an intermediary such as a financial institution).
</xsd:documentation>
             </xsd:annotation>
            </xsd:element>
            <xsd:choice>
             <xsd:element name="PayerAgentOrIntermediary" type="Party_Type">
                <xsd:annotation>
                    <xsd:documentation>Payer - Agent or Intermediary party of stfdirect. This is a party through which the actual payer has effected the payment.</xsd:documentation>
                </xsd:annotation>
             </xsd:element>
             <xsd:sequence>
                <xsd:element name="ActualPayer" type="Party_Type">
                    <xsd:annotation>
                     <xsd:documentation>Actual Payer party of stfdirect. This is the source of the payment that is described in this STF-Direct element, as determined by or documented to the sending country.

</xsd:documentation>
                    </xsd:annotation>
                </xsd:element>
                <xsd:element name="PayerAgentOrIntermediary" type="Party_Type" minOccurs="0">
                    <xsd:annotation>
                     <xsd:documentation>Payer - Agent or Intermediary party of stfdirect. This is a party through which the actual payer has effected the payment.</xsd:documentation>
                    </xsd:annotation>
                </xsd:element>
             </xsd:sequence>
            </xsd:choice>
            <xsd:element name="PaymentData" type="PaymentData_Type">
             <xsd:annotation>
                <xsd:documentation>The information about the payment that this STF-Direct element is to communicate to the receiving country. The term "payment" embodies the concept of the legal obligation to put funds at the disposal of the recipient beneficial owner of the income in the manner required by contract or by custom ("constructive receipt of income"). Therefore the interpretation of this term should not be restricted to the actual physical payment of the income (cf.:comment on Model Tax Convention Art.10 and Art.11)
</xsd:documentation>
```

```
            </xsd:annotation>
            </xsd:element>
            <xsd:element name="OtherInfo" type="OtherInfo_Type">
             <xsd:annotation>
                <xsd:documentation>Any other information regarding this STF-Direct element, e.g. errors when transforming from SMF, Filler info from SMF-Records</xsd:documentation>
             </xsd:annotation>
            </xsd:element>
        </xsd:sequence>
        <xsd:attribute name="version" fixed="1.0"/>
    </xsd:complexType>
    <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>
</xsd:schema>
```

Collection of general STF datatypes: Schema stftypes

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="urn:oecd:ties:stf:v1" xmlns="urn:oecd:ties:stf:v1" xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xsd:include schemaLocation="isotypes_v1.xsd"/>
    <xsd:include schemaLocation="oecdtypes_v1.xsd"/>
    <!-- -->
    <!-- Simple Types for the Family of OECD STF documents ___ in alphabetical order -->
    <!-- -->
    <!-- -->
    <!-- Technical Indication for Handling of the Document -->
    <xsd:simpleType name="DocTypeIndic_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">This element is used to indicate whether the data in this document part of the message is new -1-, a replication of data already sent (possibly not received) -0- or corrections to data transmitted before -2-. The element applies only to the document part in which it is included. In the case of repeated or corrected data elements CorrMessageRefId and CorrDocRefId must contain the identifiers MessageRefId and DocRefId respectively for the data referred to. Whenever the data referenced by a replication or correction is not found, the transmitted data shall be treated as new. Documents shall be transmitted and, what is even more important, processed in the following order: repeated - new - correction. In the case of a correction  the unchanged elements shall be transmitted again (i.e., repeated) - except for the element DocRefId.

</xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:byte">
            <xsd:enumeration value="0"/>
            <xsd:enumeration value="1"/>
            <xsd:enumeration value="2"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!-- -->
    <!-- Code for the gender of an individual person -->
    <xsd:simpleType name="Gender_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            This element can be used to indicate Gender for individuals. Unlike the SMF, this field is optional and can be omitted if the gender is unknown. There is no need to provide a gender element for non-individuals, as these are described by another data type.</xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="M"/>
            <xsd:enumeration value="F"/>
```

```
        </xsd:restriction>
    </xsd:simpleType>
    <!--  International Bank Account Number -->
    <xsd:simpleType name="IBAN_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            The International Bank Account Number has to be given here for the account into which the payment in question
has been made. Depending on the transmission type this element is optional. Its structure is:
Country code, 2 letters/Check digits, 2 digits/Basic Bank Account Number (BBAN), 10  to 30 alphanumeric characters
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="[A-Z]{2}[0-9]{2}[0-9,A-Z]{10,30}"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!--  -->
    <!--  International Securities Identification Number  -->
    <xsd:simpleType name="ISIN_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            The International Securities Identification Number relevant to the reported payment. Its structure is:
Country code, 2 letters/Main code, 9 alphanumeric characters/Check digit, 1 digit
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="[A-Z]{2}[0-9,A-Z]{9}[0-9]{1}"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!--  -->
    <!--  Type of the address considered from a legal point of view -->
    <xsd:simpleType name="legalAddressType_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            This is a datatype for an attribute to an address. It serves to indicate the legal character of that address
(residential, business etc.)
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="residentialOrBusiness"/>
            <xsd:enumeration value="residential"/>
            <xsd:enumeration value="business"/>
            <xsd:enumeration value="registeredOffice"/>
            <xsd:enumeration value="unspecified"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!--  -->
    <!--  Kind of Name  -->
    <xsd:simpleType name="nameType_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
             It is possible for stf documents to contain several names for the same party. This is a qualifier to indicate the type
of a particular name.  Such types include nicknames ('nick'), names under which a party does business ('dba' a short name for
the entity, or a name that is used for public acquaintance instead of the official business name) etc. Examples: in the United
States, DaimlerChrysler is still known simply as Chrysler, Dr. William Black dba Quality Pedriatics, Inc. 'SMFAliasOrOther'
should be chosen if the document is generated from a legacy SMF record, where no further distinction is possible.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <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"/>
```

```
        </xsd:restriction>
    </xsd:simpleType>
    <!--  -->
    <!--  Kind of the Identifier that is provided for a party  -->
    <xsd:simpleType name="partyIdType_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            This is to designate the kind of the identifier that is provided for a party. The party can be identified by a variety of identification numbers, codes etc. Preferably the partyIdType should be a TIN, nevertheless in the absence of a TIN other identifiers may be helpful, such as a tax file number -TFN. The element 'PartyId' that has an attribute 'partyIdType' of type 'partyIdType_Type' has another attribute to indicate the body that has issued the identifier.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="TIN"/>
            <xsd:enumeration value="IdNo"/>
            <xsd:enumeration value="TFN"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!--  -->
    <!--  Kind of the Qualifier that describes a Payment  Type-->
    <xsd:simpleType name="paymentTypeQlf_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            Kind of the qualifier that describes a payment  type. A 'Payment' element is accompanied by up to two elements 'PaymentType' for the indication of the payment's type. 'paymentTypeQlf' is an attribute of 'PaymentType' indicating the codelist this payment type code is taken from.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="opt">
             <xsd:annotation>
                <xsd:documentation xml:lang="en">OECD  payment type</xsd:documentation>
             </xsd:annotation>
            </xsd:enumeration>
            <xsd:enumeration value="cpt">
             <xsd:annotation>
                <xsd:documentation xml:lang="en">country specific payment type</xsd:documentation>
             </xsd:annotation>
            </xsd:enumeration>
            <xsd:enumeration value="sd1">
             <xsd:annotation>
                <xsd:documentation xml:lang="en">EU savings directive payment type kind 1</xsd:documentation>
             </xsd:annotation>
            </xsd:enumeration>
            <xsd:enumeration value="sd2">
             <xsd:annotation>
                <xsd:documentation xml:lang="en">EU savings directive payment type kind 2</xsd:documentation>
             </xsd:annotation>
            </xsd:enumeration>
        </xsd:restriction>
    </xsd:simpleType>
    <!--  -->
    <!--  Qualifier for a Payment:  Gross or Net Income, Tax Withheld or Refunded -->
    <xsd:simpleType name="paymentQlf_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            Qualifier for a Payment:  Gross or Net Income, Tax Withheld or Refunded
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="gip"/>
            <xsd:enumeration value="nip"/>
            <xsd:enumeration value="twh"/>
            <xsd:enumeration value="trf"/>
        </xsd:restriction>
```

```
    </xsd:simpleType>
    <!--  -->
    <xsd:simpleType name="SWIFT_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            Registration Authority for the Bank Identifier Code:
            Bank code, 4 alphanumeric characters/Country code, 2 letters/Location code, 2 alphanumeric characters /Branch code, 3 alphanumeric characters, optional
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="[0-9,A-Z]{4}[A-Z]{2}[0-9,A-Z]{2}([0-9,A-Z]{3})?"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!--  -->
    <!-- A list of entries to mark the tax years relevant to the document (part). The years are in the form of dates denoting the last day of the tax year in the respective country -->
    <xsd:simpleType name="TaxYearList_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            A list of entries to mark the tax years relevant to the document (part). The years are in the form of dates denoting the last day of the tax year in the respective country
            </xsd:documentation>
        </xsd:annotation>
        <xsd:list itemType="xsd:date"/>
    </xsd:simpleType>
    <!--  -->
    <!-- Data type for tax rates -->
    <xsd:simpleType name="TaxRate_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            Data type for tax rates. Tax rates have to be entered as decimal numbers with a total of four digits, two before and two after the decimal point.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:decimal">
            <xsd:totalDigits value="4"/>
            <xsd:fractionDigits value="2"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!--  -->
    <!-- Data type for any kind of numeric data with two decimal fraction digits, especially monetary amounts -->
    <xsd:simpleType name="TwoDigFract_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            Data type for any kind of numeric data with two decimal fraction digits, especially monetary amounts
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:decimal">
            <xsd:fractionDigits value="2"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!--  -->
    <!--  -->
    <!--  Complex Types for the Family of OECD STF documents  -->
    <!--  here: Complex Types of a General Kind  ___  in alphabetical order -->
    <!--  Types that are specific for a particular message type will be defined in the schema of that message document -->
    <!--  -->
    <!-- Description of an Account  -->
    <xsd:complexType name="AcctInfo_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            It may be desirable or even necessary to extend the information about a payment by including information about the account that was used for the payment and/or the security to which the payment relates. One or more such entries can be given in an element of this type. The element itself is optional unless stated otherwise for a particular document type, if it is present, however, it must not be empty.
```

```
            </xsd:documentation>
        </xsd:annotation>
        <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>
    <!--  -->
    <!-- Structure of the Address of a Party broken down into its logical Parts -->
    <xsd:complexType name="AddressFix_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            Structure of the address for a party broken down into its logical parts, recommended for easy matching. The 'City'
element is the only required subelement. All of the subelements are simple text - data type 'string'.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="Street" type="xsd:string" minOccurs="0"/>
            <xsd:element name="BuildingIdentifier" type="xsd:string" minOccurs="0"/>
            <xsd:element name="SuiteIdentifier" type="xsd:string" minOccurs="0"/>
            <xsd:element name="FloorIdentifier" type="xsd:string" minOccurs="0"/>
            <xsd:element name="DistrictName" type="xsd:string" minOccurs="0"/>
            <xsd:element name="POB" type="xsd:string" minOccurs="0"/>
            <xsd:element name="PostCode" type="xsd:string" minOccurs="0"/>
            <xsd:element name="City" type="xsd:string"/>
            <xsd:element name="CountrySubentity" type="xsd:string" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <!--  -->
    <!--  The Address of a Party, given in fixed or free Form, possibly in both Forms -->
    <xsd:complexType name="Address_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            The user has the option to enter the data about the address of a party either as one long field or to spread the data
over up to eight  elements or even to use both formats. If the user chooses the option to enter the data required in separate
elements, the container element for this will be 'AddressFix'. If the user chooses the option to enter the data required in a less
structured way in 'AddressFree' all available address details shall be presented as one string of bytes, blank or "/" (slash) or
carriage return- line feed used as a delimiter between parts of the address. PLEASE NOTE that the address country code is
outside  both of these elements. The use of the fixed form is recommended as a rule to allow easy matching. However, the
use of the free form is recommended if the sending state cannot reliably identify and distinguish the different parts of the
address. The user may want to use both formats e.g. if besides separating the logical parts of the address he also wants to
indicate a suitable breakdown into print-lines by delimiters in the free text form. in this case 'AddressFix' has to precede
'AddressFree'.


            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="CountryCode" type="CountryCode_Type"/>
            <xsd:choice>
             <xsd:element name="AddressFree" type="xsd:string"/>
             <xsd:sequence>
                 <xsd:element name="AddressFix" type="AddressFix_Type"/>
                 <xsd:element name="AddressFree" type="xsd:string" minOccurs="0"/>
             </xsd:sequence>
            </xsd:choice>
        </xsd:sequence>
        <xsd:attribute name="legalAddressType" type="legalAddressType_Type"/>
    </xsd:complexType>
    <!--  -->
    <!--  Document specification: Data identifying and describing the document -->
```

```
<xsd:complexType name="DocSpec_Type">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">Document specification: Data identifying and describing the document, where 'document' here means the part of a message that is to transmit the data about a single payment/transaction or other meaningful self-contained chunk of information. STF messages do not factorize such information in order to transmit repeating data only once (e.g. data about a party that has received multiple payments). 'DocRefId' is an identifier that the sender has to attribute to this document and which has to be unique at least inside the containing message. If the document refers to another one transmitted before, 'CorrMessageRefId' and 'CorrDocRefId' have to contain the  corresponding Id's of the message and document referred to.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="DocTypeIndic" type="DocTypeIndic_Type"/>
        <xsd:element name="DocRefId" type="xsd:string">
         <xsd:annotation>
            <xsd:documentation xml:lang="en">Sender's unique identifier of this document </xsd:documentation>
         </xsd:annotation>
        </xsd:element>
        <xsd:element name="CorrMessageRefId" type="xsd:string" minOccurs="0">
         <xsd:annotation>
            <xsd:documentation  xml:lang="en">Reference id of the message of the document referred to if this is repetition or correction</xsd:documentation>
         </xsd:annotation>
        </xsd:element>
        <xsd:element name="CorrDocRefId" type="xsd:string" minOccurs="0">
         <xsd:annotation>
            <xsd:documentation  xml:lang="en">Reference id of the document referred to if this is repetition or correction</xsd:documentation>
         </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<!--  -->
<!-- Data (other than Name and Address) to describe and identify an Individual  -->
<xsd:complexType name="IndivPersData_Type">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
        Data (other than Name and Address) to describe and identify an Individual. In general all of the subelements are optional, within certain document types they may be made obligatory. 'Nationality' is of type 'CountryCode_Type' --&gt;, 'BirthDate' is of type date, that is in the  form ccyy-mm-dd, the content of all other subelements is character string.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="Gender" type="Gender_Type" minOccurs="0"/>
        <xsd:element name="Nationality" type="CountryCode_Type" minOccurs="0"/>
        <xsd:element name="BirthDate" type="xsd:date" minOccurs="0"/>
        <xsd:element name="BirthCity" type="xsd:string" minOccurs="0"/>
        <xsd:element name="BirthCitySubentity" type="xsd:string" minOccurs="0"/>
        <xsd:element name="BirthCountryCode" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
<!--  -->
<!-- Data (other than Name and Address) to describe and identify a Legal Entity -->
<xsd:complexType name="LegalPersData_Type">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
        Data (other than Name and Address) to describe and identify a legal entity. Currently the foundation date is the only subelement. It is defined as required inside 'LegalPersData_Type', as an empty element should not appear in the document. However, the  element containing LegalPersData is optional. 'FoundDate' is of type date, that is in the  form ccyy-mm-dd.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="FoundDate" type="xsd:date"/>
    </xsd:sequence>
</xsd:complexType>
<!--  -->
```

```
<!-- Message specification: Data identifying and describing the message as a whole -->
<xsd:complexType name="MessageSpec_Type">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">Message specification: Data identifying and describing the message as a
whole. 'SendingCountry' and 'ReceivingCountry' are to identify the relation of the transmission, so that this is visible
independently of the transmission context inside and at the very top of the message. The elements are optional as in the SMF
record there are no fields exactly corresponding; it is, however, strongly recommended to use these fields as intended.
'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 given. '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 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.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="SendingCountry" type="CountryCode_Type" minOccurs="0"/>
        <xsd:element name="ReceivingCountry" type="CountryCode_Type" minOccurs="0"/>
        <xsd:element name="Warning" type="xsd:string">
         <xsd:annotation>
            <xsd:documentation xml:lang="en">Free text expressing the restrictions for use of the information this
message contains and the legal framework under which it is given</xsd:documentation>
         </xsd:annotation>
        </xsd:element>
        <xsd:element name="Contact" type="xsd:string">
         <xsd:annotation>
            <xsd:documentation xml:lang="en">All necessary contact information about persons responsible for and
involved in the processing of the data transmitted in this message, both legally and technically. Free text as this is not
intended for automatic processing. </xsd:documentation>
         </xsd:annotation>
        </xsd:element>
        <xsd:element name="MessageRefId" type="xsd:string">
         <xsd:annotation>
            <xsd:documentation xml:lang="en">Sender's unique identifier for this message</xsd:documentation>
         </xsd:annotation>
        </xsd:element>
        <xsd:element name="TaxYearList" type="TaxYearList_Type">
         <xsd:annotation>
            <xsd:documentation xml:lang="en">A list of all tax years for which information is transmitted in documents of
the current message. To indicate a tax year, the date of the last day of that year is given.</xsd:documentation>
         </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<!-- -->
<!-- General Type for Monetary Amounts -->
<xsd:complexType name="MonAmnt_Type">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
        This data type is to be used whenever monetary amounts are to be communicated. Such amounts shall be given
including two fractional digits of the main currency unit. The code for the currency in which the value is expressed has to be
taken from the ISO codelist 4217 and added in attribute currCode.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
        <xsd:extension base="TwoDigFract_Type">
         <xsd:attribute name="currCode" type="currCode_Type" use="required"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
<!-- Structure of the Name of a Party broken down into its logical Parts -->
<xsd:complexType name="NameFix_Type">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
```

Structure of the name of a party broken down into its logical parts, recommended for easy matching. This type is constructed following the PersonName complex data type of the OASIS CIQ xNL standard. To keep STF as simple as possible it is not formally constructed as a xsd:restriction of that type.

```
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="PrecedingTitle" minOccurs="0" maxOccurs="unbounded">
             <xsd:annotation>
                 <xsd:documentation>His Excellency,Estate of the Late ...</xsd:documentation>
             </xsd:annotation>
             <xsd:complexType mixed="true"/>
            </xsd:element>
            <xsd:element name="Title" minOccurs="0" maxOccurs="unbounded">
             <xsd:annotation>
                 <xsd:documentation>Greeting title. Example: Mr, Dr, Ms, Herr, etc. Can have multiple
titles.</xsd:documentation>
             </xsd:annotation>
             <xsd:complexType mixed="true"/>
            </xsd:element>
            <xsd:element name="FirstName" minOccurs="0" maxOccurs="unbounded">
             <xsd:annotation>
                 <xsd:documentation>Represents the position of the name in a name string. Can be Given Name, Forename,
Christian Name, Surname, Family Name, etc. Use the attribute "NameType" to define what type this name
is.</xsd:documentation>
             </xsd:annotation>
             <xsd:complexType mixed="true">
                 <xsd:attribute name="xnlNameType">
                     <xsd:annotation>
                      <xsd:documentation>Defines the name type of FirstName. Example: Given Name, Forename, Christian
Name, Father's Name, etc. In some countries, FirstName could be a Family Name or a SurName. Use this attribute to define
the type for this name.</xsd:documentation>
                     </xsd:annotation>
                 </xsd:attribute>
             </xsd:complexType>
            </xsd:element>
            <xsd:element name="MiddleName" minOccurs="0" maxOccurs="unbounded">
             <xsd:annotation>
                 <xsd:documentation>Middle name (essential part of the name for many nationalities). Represents the position
of the name in the name string. Example: Sakthi in "Nivetha Sakthi Shantha". Can have multiple middle
names.</xsd:documentation>
             </xsd:annotation>
             <xsd:complexType mixed="true">
                 <xsd:attribute name="xnlNameType">
                     <xsd:annotation>
                      <xsd:documentation>Defines the name type of Middle Name. Example: First name, middle name,
maiden name, father's name, given name, etc.</xsd:documentation>
                     </xsd:annotation>
                 </xsd:attribute>
             </xsd:complexType>
            </xsd:element>
            <xsd:element name="NamePrefix" minOccurs="0">
             <xsd:annotation>
                 <xsd:documentation>de, van, van de, von, etc. Example: Derick de Clarke</xsd:documentation>
             </xsd:annotation>
             <xsd:complexType mixed="true">
                 <xsd:attribute name="xnlNameType">
                     <xsd:annotation>
                      <xsd:documentation>Defines the type of name associated with the NamePrefix. For example the type of
name is LastName and this prefix is the prefix for this last name.</xsd:documentation>
                     </xsd:annotation>
                 </xsd:attribute>
             </xsd:complexType>
            </xsd:element>
            <xsd:element name="LastName" minOccurs="0" maxOccurs="unbounded">
             <xsd:annotation>
```

```
                <xsd:documentation>Represents the position of the name in a name string. Can be Given Name, Forename, Christian Name, Surname, Family Name, etc. Use the attribute "NameType" to define what type this name is.</xsd:documentation>
            </xsd:annotation>
            <xsd:complexType mixed="true">
                <xsd:attribute name="xnlNameType">
                    <xsd:annotation>
                     <xsd:documentation>Defines the name type of LastName. Example: Father's name, Family name, Sur Name, Mother's Name, etc. In some countries, LastName could be the given name or first name.</xsd:documentation>
                    </xsd:annotation>
                </xsd:attribute>
            </xsd:complexType>
           </xsd:element>
           <xsd:element name="GenerationIdentifier" minOccurs="0" maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation>Jnr, Thr Third, III</xsd:documentation>
            </xsd:annotation>
            <xsd:complexType mixed="true"/>
           </xsd:element>
           <xsd:element name="Suffix" minOccurs="0" maxOccurs="unbounded">
            <xsd:annotation>
                <xsd:documentation>Could be compressed initials - PhD, VC, QC</xsd:documentation>
            </xsd:annotation>
            <xsd:complexType mixed="true"/>
           </xsd:element>
           <xsd:element name="GeneralSuffix" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Deceased, Retired ...</xsd:documentation>
            </xsd:annotation>
            <xsd:complexType mixed="true"/>
           </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
    <!--      The Name of a Party, given in fixed or free Form, possibly in both Forms -->
    <xsd:complexType name="Name_Type">
        <xsd:annotation>
           <xsd:documentation xml:lang="en">
           The user has the option to enter the data about the name of a party either as one long field or to spread the data over up to six elements or even to use both formats. If the user chooses the option to enter the data required in separate elements, the container element for this will be 'NameFix'. If the user chooses the option to enter the data required in a less structured way in 'NameFree' all available details on the name of the party shall be presented as one string of bytes, blank or "/" (slash) used as a delimiter between parts of the name.  The use of the fixed form is recommended as a rule to allow easy matching. However, the use of the free form is recommended if the sending state cannot reliably identify and distinguish the different parts of the name. The user may want to use both formats in special circumstances. In this case 'NameFix' has to precede 'NameFree'.
An optional attribute 'nameType' can be used to indicate a special kind of name, as for instance a nickname, a name-at-birth etc.
           </xsd:documentation>
        </xsd:annotation>
        <xsd:choice>
           <xsd:element name="NameFree" type="xsd:string"/>
           <xsd:sequence>
            <xsd:element name="NameFix" type="NameFix_Type"/>
            <xsd:element name="NameFree" type="xsd:string" minOccurs="0"/>
           </xsd:sequence>
        </xsd:choice>
        <xsd:attribute name="nameType" type="nameType_Type" use="optional"/>
    </xsd:complexType>
    <!--  -->
    <!--  Other Bank Account Number: A Bank Account Number other than the standard IBAN, the attribute to indicate the kind of such number  -->
    <xsd:complexType name="OBAN_Type">
        <xsd:annotation>
           <xsd:documentation xml:lang="en">
           Other Bank Account Number: A bank account number other than the standard IBAN, the attribute 'acctNoQlf' has to be used to indicate the kind of such number.
```

```
            </xsd:documentation>
        </xsd:annotation>
        <xsd:simpleContent>
            <xsd:extension base="xsd:string">
             <xsd:attribute name="acctNoQlf" type="xsd:string" use="required"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <!--  -->
    <!--  Other Security Identification Number: A Security Identification Number other than the standard ISIN, the attribute to indicate the kind of such number  -->
    <xsd:complexType name="OSIN_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            Other Security Identification Number: A security identification number other than the standard ISIN, the attribute 'secNoQlf' has to be used to indicate the kind of such number.

            </xsd:documentation>
        </xsd:annotation>
        <xsd:simpleContent>
            <xsd:extension base="xsd:string">
             <xsd:attribute name="secNoQlf" type="xsd:string" use="required"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <!-- A "Filler" type to accommodate any additional information elements a sender might want to add in order to enhance the value of the "standard" content -->
    <xsd:complexType name="OtherInfo_Type" mixed="true">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            OtherInfo is an element to accommodate any additional information that a sender might want to add in order to enhance the value of the "standard" content. The sender has to make sure both by using adequate tag names and adding explanations that the receiver is able to understand 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.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:complexContent mixed="true">
            <xsd:restriction base="xsd:anyType">
             <xsd:sequence>
                 <xsd:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
             </xsd:sequence>
            </xsd:restriction>
        </xsd:complexContent>
    </xsd:complexType>
    <!-- Identifier for a Party -->
    <xsd:complexType name="PartyId_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            This is the type of an element 'PartyId' which is to contain an identification number/identification code for the party in question. As the identifier may be not strictly numeric, it is just defined as a string of characters. Attributes 'partyIdType' and 'issuedBy' are required to designate the kind (e.g. TIN) and issuer of the identifier. In the case of a TIN the issuer attribute has to be the ISO country code of the issuing country. This has to be guaranteed by the sender without  the type of issuedBy being formally restricted to CountryCode_Type. (In non-TIN cases issuedBy may have to contain some information the kind of which is not known in advance, so as to the formal typing we have to stay here somewhat ambiguous.)
            </xsd:documentation>
        </xsd:annotation>
        <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>
    <!--  Collection of all Data describing a Party -->
    <xsd:complexType name="Party_Type">
        <xsd:annotation>
```

```
            <xsd:documentation xml:lang="en">
            This container brings together all data about a party. Name and address are required components and each can
be present more than once to enable as complete a description as possible. Whenever possible one or more identifiers (TIN
etc) should be added as well as a residence country code.  Additional data that describes and identifies the party can be
given in the 'PersData' element. The code for the legal type according to the OECD codelist must be added.  The structures of
all of the subelements are defined elsewhere in this schema.</xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="ResCountryCode" type="CountryCode_Type" minOccurs="0"/>
            <xsd:element name="PartyId" type="PartyId_Type" minOccurs="0" maxOccurs="3"/>
            <xsd:element name="Name" type="Name_Type" maxOccurs="unbounded"/>
            <xsd:element name="Address" type="Address_Type" maxOccurs="unbounded"/>
            <xsd:element name="PersData" type="PersData_Type" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="oecdLegalType" type="oecdLegalType_Type" use="required"/>
    </xsd:complexType>
    <!--  -->
    <!-- Kind of the Payment -->
    <xsd:complexType name="PaymentType_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            A code has to be entered here to describe the nature of the payment. As this code can be taken from a variety of
codelists, the attribute 'paymentTypeQlf' --&gt; will indicate the relevant codelist and may itself be qualified by another -
optional - attribute 'paymentTypeQlfQlf'. Example: A paymentTypeQlf value of 'cpt' indicates that the value for PaymentType
has been taken from a country specific codelist; in this case paymentTypeQlfQlf should give details about that codelist, e.g.
the issuing country. As the data type for paymentTypeQlfQlf is just "string", there is no restriction on the format of the
information contained in this element. If paymentTypeQlf has a value of 'opt' the content of the element has to be not only of
type xsd:string but of oecdPaymentType_Type; this, however, is not reflected - nor enforced - by this schema.

            </xsd:documentation>
        </xsd:annotation>
        <xsd:simpleContent>
            <xsd:extension base="xsd:string">
             <xsd:attribute name="paymentTypeQlf" type="paymentTypeQlf_Type" use="required"/>
             <xsd:attribute name="paymentTypeQlfQlf" type="xsd:string" use="optional"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <!-- Data (other than Name and Address) to describe and identify a Party  -->
    <xsd:complexType name="PersData_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            Data (other than Name and Address) to describe and identify a party. Depending on the type of the party
(individual or legal person) element 'IndivPersData' or element 'LegalPersData' must be used.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:choice>
            <xsd:element name="IndivPersData" type="IndivPersData_Type"/>
            <xsd:element name="LegalPersData" type="LegalPersData_Type"/>
        </xsd:choice>
    </xsd:complexType>
    <!--  -->
    <!-- Collection of all Data describing a Payment  -->
    <xsd:complexType name="PaymentData_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">Payment Data within international data exchange for tax purposes.  In this
container all payment data is brought together. The element 'PaymentType' is required, can occur twice and describes the
reason for the payment (e.g. interest). It may use different terms (OECD codelist vs. country specific codelist). This type can
involve several  "Payment" elements, which represent part of a group  (gross payment, net payment, tax deducted etc.). To
indicate the relevant tax year, the date of the last day of that year is given.  In a PaymentData element as a rule the following
equations should hold between the amounts in the Payment child elements that are distinguished by the paymentQlf attribute:
NIP = GIP - TWH; TWH = GIP * TR (the paymentQlf values are used in these equations to identify the amounts between
which the equations hold) . Amounts that can be calculated from the others by virtue of these equations do not necessarily
have to be entered. If amounts are entered for which the above equations do not hold, an explanation should be provided in
the OtherInfo element. It is assumed that tax refund is not bound to the other amounts by an equation of the above kind.
</xsd:documentation>
```

```
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="TaxYearEnd" type="xsd:date">
             <xsd:annotation>
                 <xsd:documentation xml:lang="en">To indicate the relevant tax year, the date of the last day of that year is given.</xsd:documentation>
             </xsd:annotation>
            </xsd:element>
            <xsd:element name="PaymentType" type="PaymentType_Type" maxOccurs="2"/>
            <xsd:element name="Payment" type="Payment_Type" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <!--  -->
    <!-- A single Item describing one Aspect of a Payment, e.g. a Net Payment, a Tax Payment -->
    <xsd:complexType name="Payment_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            A single item describing one aspect of a payment, e.g. a net payment, a tax payment - distinguished by the required attribute 'paymentQlf'. The payment should be described as precisely as possible, even if all of the subelements except the monitary amount itself are optional.
            </xsd:documentation>
        </xsd:annotation>
        <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>
</xsd:schema>
```

Codelist Schemas (for ISO codes only an extract is shown)

isotypes_v1.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
    <!--  ISO 3166 alpha 2 Country Code -->
    <xsd:simpleType name="CountryCode_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            The appropriate country code from the ISO 3166 two-byte alpha version for the state of which the party concerned is a resident. Omit this only if  no data is available.
Valid entries are:
    - AF --  AFGHANISTAN
    - AL --  ALBANIA
    ...
    - ZM --  ZAMBIA
    - ZW --  ZIMBABWE
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="AF"/>
            <xsd:enumeration value="AL"/>
            ...
            <xsd:enumeration value="ZM"/>
            <xsd:enumeration value="ZW"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!--  ISO 4217 alpha 3 Currency Code -->
    <xsd:simpleType name="currCode_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            The appropriate currency code from the ISO 4217 three-byte alpha version for the currency in which a monetary amount is expressed.
Valid entries are:
```

```
AED United Arab Emirates, Dirhams
AFA Afghanistan, Afghanis
...
ZMK Zambia, Kwacha
ZWD Zimbabwe, Zimbabwe Dollars
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="AED"/>
            <xsd:enumeration value="AFA"/>
            ...
            <xsd:enumeration value="ZMK"/>
            <xsd:enumeration value="ZWD"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
```

oecdtypes_v1.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xsd:simpleType name="oecdLegalType_Type">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            The OECD code describing the legal status of the party:
            01 Individual
            02 Corporation
            03 Partnership
            04 Business organisation other than corporation or partnership
            05 Government or international organisation
            06 Other (specify in the 'OtherInfo' element)
            07 Unknown
            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="01"/>
            <xsd:enumeration value="02"/>
            <xsd:enumeration value="03"/>
            <xsd:enumeration value="04"/>
            <xsd:enumeration value="05"/>
            <xsd:enumeration value="06"/>
            <xsd:enumeration value="07"/>
        </xsd:restriction>
    </xsd:simpleType>
    <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
            07 Business profits
            10 Dividends
            11 Interest
            12 Royalties
            13 Capital Gains
            14 Income from independent personal services
            15 Income from dependent personal services
            15a Gross amount (including fringe benefits)
            15b Money amount only but additional information on fringe benefits to follow in fillers according to bilateral
agreements
            15c Money amount only
            16 Directors' fees
            17 Income derived from activities of an artist or sportsman
            18 Pensions
            19 Income from government services and public pensions
            20 Payments to students for education and training
```

```
            21 Other income

            </xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="06"/>
            <xsd:enumeration value="07"/>
            <xsd:enumeration value="10"/>
            <xsd:enumeration value="11"/>
            <xsd:enumeration value="12"/>
            <xsd:enumeration value="13"/>
            <xsd:enumeration value="14"/>
            <xsd:enumeration value="15"/>
            <xsd:enumeration value="15a"/>
            <xsd:enumeration value="15b"/>
            <xsd:enumeration value="15c"/>
            <xsd:enumeration value="16"/>
            <xsd:enumeration value="17"/>
            <xsd:enumeration value="18"/>
            <xsd:enumeration value="19"/>
            <xsd:enumeration value="20"/>
            <xsd:enumeration value="21"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
```