CENTRE FOR TAX POLICY AND ADMINISTRATION
COMMITTEE ON FISCAL AFFAIRS

OECD MANUAL ON THE IMPLEMENTATION OF EXCHANGE OF INFORMATION FOR TAX PURPOSES - AUTOMATIC (OR ROUTINE) EXCHANGE OF INFORMATION

Module 3 - Automatic (or Routine) Exchange of Information

This document was approved by delegates to Working Party No.8.

The Manual is now submitted to the CFA for APPROVAL under the written procedure. Please submit your comments no later than January 9, 2006 to Ms. Suzanne Pedron (suzanne.pedron@oecd.org). If you can approve the Manual as it is you do not need to respond and the absence of comments will be taken to mean approval.

For further information please contact: Mrs. Martine Milliet-Einbinder: Tel: (33 1) 45 24 78 42; Fax: (33 1) 44 30 63 21; Email: martine.milliet-einbinder@oecd.org
# TABLE OF CONTENTS

## OECD MANUAL ON THE IMPLEMENTATION OF EXCHANGE OF INFORMATION FOR TAX PURPOSES

1. What is automatic exchange? ........................................................................................................... 3
2. Benefits of automatic exchange........................................................................................................ 3
3. Legal basis ........................................................................................................................................ 3
4. Agreements or Memoranda of Understanding on automatic exchange ............................................ 4
5. Implementation ................................................................................................................................. 5
   5.1 Standardisation of transmission formats and use of new media ................................................. 5
   5.2 Security: Encryption and alternative methods ............................................................................ 6
6. Importance of feedback from receiving country .............................................................................. 6
7. Future developments: Risk analysis tools ........................................................................................ 6

## ANNEXES (ALSO AVAILABLE ON THE SECURE WEBSITE FOR EXCHANGE OF INFORMATION STAFF)

1. 1997 REVISED OECD STANDARD MAGNETIC FORMAT FOR AUTOMATIC EXCHANGE OF INFORMATION .......................................................... 7
2. USER GUIDE FOR OECD STANDARD MAGNETIC FORMAT (SMF) ................................................ 23
3. OECD STANDARISED TRANSMISSION FORMAT FOR AUTOMATIC EXCHANGE OF INFORMATION FOR TAX PURPOSES.............................................. 60
4. USER GUIDE FOR THE OECD STANDARD TRANSMISSION FORMAT........................................ 78
5. BRIDGING PROGRAMME FROM SMF TO STF ................................................................................. 101
6. BRIDGING PROGRAMME FROM STF TO SMF ............................................................................... 111
1. **What is automatic exchange?**

   Automatic exchange of information (also called routine exchange by some countries) involves the systematic and periodic transmission of “bulk” taxpayer information by the source country to the residence country concerning various categories of income (e.g. dividends, interest, royalties, salaries, pensions, etc). This information is obtained on a routine basis in the source country (generally through reporting of the payments by the payer (financial institution, employer etc). Automatic exchange can also be used to transmit other useful types of information such as changes of residence, the purchase or disposition of immovable property, VAT refunds\(^1\), etc. The country of residence tax authority can then check its tax records to verify that resident taxpayers have reported their foreign source income. In addition, information concerning the acquisition of significant assets may be used to evaluate the net worth of an individual, to see if the reported income reasonably supports the transaction. There are an increasing number of countries involved in automatic exchange using different types of media: tapes, diskettes, CD Roms but also paper.

2. **Benefits of automatic exchange**

   The foreign source information received on magnetic media or in digital form can be input into the recipient tax data base (often using bridging programs to capture the relevant information) and automatically matched against the income reported by the taxpayer. This is the most cost effective way to process the information. For example the Australian Tax Office's 2004-05 Compliance Program states that 1171 foreign source income data matching audits were completed during the 2003-04 tax year, raising over AUD$3 million in liabilities. The foreign source information received on magnetic media or in digital form can also be matched manually, as a general procedure or when it could not be matched automatically. The automatic exchange of information on magnetic media also provides opportunities for more effective and efficient distribution of the exchanged information to local tax offices if needed and also for instance for feeding the information into data bases for purposes of risk analysis.

3. **Legal basis**

   Automatic exchange can be based on:

   i. The exchange of information article of the bilateral income tax convention between two countries; or

\(^1\) These other types are not at present supported by the standard transmission formats (SMF/STF) developed by the OECD.
ii. Article 6 of the Joint Council of Europe/OECD Convention on Mutual Administrative Assistance in Tax Matters; or

iii. Article 3 of the EU Council Directive 77/799/EEC on Mutual Assistance as last amended or

iv. The EU Savings Directive 2003/48/EC; or

v. Article 17 of EU Council Regulation on administrative cooperation in the field of VAT 1798/2003; or

vi. Council Regulation of 16 November 2004 on administrative co-operation in the field of excise due; or

vii. Article 4, paragraph 3 of the CIAT Model Agreement on the Exchange of Tax Information.

4. **Agreements or Memoranda of Understanding on automatic exchange**

4. The exchange of information article of an income tax convention or automatic exchange article of a mutual assistance instrument constitutes the legal basis for automatic exchange of information. In addition, countries may agree to enter into a special working agreement or memorandum of understanding (MOU) setting forth the terms and conditions of the proposed automatic exchange. Such an agreement or MOU typically sets forth the types of information to be exchanged automatically, details about the procedures of sending and receiving the information, the appropriate format to use, and provision of TINs. These agreements or MOU may be published officially and may have a deterrent effect on potential tax evaders and are usually reviewed periodically.

5. The OECD has designed a Model Memorandum of Understanding between Competent Authorities on Automatic Exchange of Information for Tax Purposes C(2001)21/FINAL that can be used as a basis for an operational working agreement between tax administrations.

6. The OECD Model MOU provides a list of information that can be exchanged automatically, including:

   - change in place of residence from one State to the other State;
   - ownership of and income from immovable property;
   - dividends;
   - interest;
   - royalties;
   - capital gains;
   - salaries, wages and other similar remuneration in respect of an employment,;
   - directors’ fees and other similar payments;
   - income derived by artists and sportsmen, pensions and other similar remuneration, salaries, wages and other similar remuneration for government services, other income such as proceeds
from gambling, other items including items on indirect taxes such as VAT/sales tax and excise duties and social security payments; and

- commissions and other similar payments.

7. The OECD Model MOU recommends using the OECD Standard Magnetic Format for automatic exchange (or any further updated format recommended by the OECD Council) as well as providing Tax Identification Numbers (TINs) if available as they facilitate the processing and matching of the information received. It might also be a helpful reference when further inquiries to the other contracting party are necessary. In that respect the OECD Council recommended the use of TINs in the international context in 1997, see C(97)29/FINAL. The OECD Council recommends “that Member countries should encourage non residents recipients of income to disclose their residence country TIN. Member countries should consider making this disclosure mandatory.” In the case of automatic exchange of information on income paid to non residents, having information on the residence country TIN would greatly facilitate the matching of information received by the residence country with the income reported by its own taxpayers. The table in the annex below gives an overview of the use of TINs for domestic purposes and international use of TINs.

5. Implementation

5.1 Standardisation of transmission formats and use of new media

8. Automatic exchange of information requires the standardisation of formats in order to be efficient. In 1981 the OECD designed a paper-based form for automatic exchange which introduced the standardisation of certain pieces of information C(81)39/FINAL. In 1992, taking advantage of technological developments, the OECD then designed the Standard Magnetic Format (SMF) for the transmission of taxpayer information on magnetic tape. Based on country experiences the SMF was revised in 1997 to further improve countries’ capacity to match information received automatically with information reported by its taxpayers. Use of the revised format was recommended by the OECD Council in 1997 (see C(97)30/FINAL). The record layout of the Standard includes fields allocated to the:

- recipient beneficial owner, his agent or intermediary, to the actual payer of the income, the payer’s agent or intermediary. For each series of fields the same pattern is followed to provide information on the TIN (both residence country TIN and source country TIN), name, alias or other name, date of birth (where applicable) and address; and

- income (tax year, date, type of payment, currency, gross and net amount, tax withheld, refund etc).

9. Fields are allocated to residence country TINs and source country TINs. The SMF is used by OECD member countries involved in automatic exchange and increasingly by non member countries. A multilingual electronic user manual is also available to provide guidance for the implementation of the Standard. It is available on CD ROM which can be obtained from the OECD Secretariat. In 2002 the European Union Council agreed on a standard format (FISC 39) for the implementation of the Savings Directive which is to be based on the OECD SMF. In 2008 EU countries will start using an STF-like format (FISC 73).

10. The OECD has also designed a “new generation” transmission format for automatic exchange to eventually replace the SMF. The new format is called the Standard Transmission Format (STF) and is
based on extensible mark-up language (XML\(^2\), a document mark-up 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, etc). A multilingual electronic STF user manual is available to provide guidance for the implementation of the STF. It is available from the OECD Secretariat or on the OECD secure site for competent authorities. As the SMF and STF will coexist for the foreseeable future, 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.

5.2 Security: Encryption and alternative methods

11. It is desirable that information contained in magnetic media exchanged automatically be transmitted in a secure manner and be encrypted whenever feasible. Such information is now transmitted on diskettes or, in particular, CD ROMs, both of which can be easily encrypted by the sending country using encryption software. A pilot has tested encryption software and a few countries now provide information on encrypted CD ROMs. The tested encryption software GNU PG\(^3\) or equivalent commercial software has been found adequate to ensure the security of automatic exchange. If not encrypted, the diskettes tapes or CD ROMs should be exchanged via a secure mail system, such as a diplomatic pouch.

6. Importance of feedback from receiving country

12. Feedback to the sending country is essential to improve the efficiency of automatic exchange of information. Feedback from the receiving country on information exchanged automatically (not purely from an IT perspective) is crucial to make better use of what is exchanged: knowing what the source is of data exchanged, the common errors identified, etc. Feedback may also be useful to tax administrations for justifying resources for exchange of information. More generally, a systematic survey of the use of the OECD standard magnetic format is carried out on a regular basis by the OECD. Feedback includes comments on the accessibility, accuracy, and completeness of the data received as well as comments on the usefulness of the data, such as summary results of taxpayer reviews and audits.

7. Future developments: Risk analysis tools

13. Countries are now starting to use foreign source information received automatically not only for matching purposes but also in a more strategic manner for risk assessment purposes, and in particular to select cases for more in depth tax examinations.

\(^2\) 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.

\(^3\) GNU Privacy Guard is a suite of programmes developed by the Free Software Foundation that provides security solutions for protecting and encrypting data (see www.gnupgp.org). It uses the Pretty Good Privacy (PGP) standard and is compatible with commercial PGP products. PGP is a widely used encryption programme designed to provide high-security encoding algorithms.
ANNEXES (ALSO AVAILABLE ON THE SECURE WEBSITE FOR EXCHANGE OF INFORMATION STAFF)

OECD Recommendation for Memorandum of Understanding on automatic exchange (see www.oecd.org/taxation)

OECD Standard Magnetic Format (SMF)

OECD Standard Transmission Format (STF)

Bridging programme from SMF to STF

Bridging programme from STF to SMF

Reference Guide on Sources of Information from Abroad (see www.oecd.org/taxation)
1997 REVISED OECD STANDARD MAGNETIC FORMAT
FOR AUTOMATIC EXCHANGE OF INFORMATION

Technical introduction

The revised OECD Standard Magnetic Format is an improved version of the 1992 Standard Magnetic Format. The revised OECD Standard Magnetic Format includes the Standard for the:

1. Record lay out and two tables
2. An OECD Foreign Data Exchange Magnetic Media Specifications
3. Physical label attached to the magnetic media

1. Record Lay out

The international standards (ISO codes) established by the International Organization for Standardization are used as codes for country names, currencies, dates, definition of characters used, file structure and labelling of the media. Two tables indicate the code for type of recipient and type of payer as well as the code for income.

The Record lay out includes 104 fields with filler fields at the end:

Fields 2 to 40 are allocated to the recipient Beneficial Owner,
Fields 41 to 55 are allocated to the recipient beneficial owner Agent or Intermediary
Fields 56 to 71 are allocated to the actual payer of the income
Fields 72 to 86 are allocated to the payer’s agent or intermediary

For each series of fields the same pattern is followed to provide information on the Tax Identification Number, name, alias or other name and address.
Fields 87 to 100 are allocated to the income (tax year, date, type of payment, currency, gross and net amount, tax withheld, refund etc.).
Fields 101 and 102 are allocated to references
Fields 103 and 104 are allocated to fillers to provide additional information

For each field are provided the numbering of the field, the starting position of the field, its length, the data type: N= numeric, AN= alphanumeric, B= blank.

A user manual will be provided to give general guidance and country specifications.
Revised OECD Standard Magnetic Format Lay out

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Start</th>
<th>Length</th>
<th>Data Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Type Indicator</td>
<td>1</td>
<td>1</td>
<td>N</td>
<td>0 - Repeat Data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 - New Data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 - Correction</td>
</tr>
<tr>
<td>Recipient Beneficial Owner Residence Country Code</td>
<td>2</td>
<td>2</td>
<td>A</td>
<td>ISO 3166</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 Alpha or Blank</td>
</tr>
<tr>
<td>Recipient Beneficial Owner Residence Country Code</td>
<td>4</td>
<td>20</td>
<td>AN</td>
<td>Residence Country Tax</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Identification Number or Blank if data not available</td>
</tr>
<tr>
<td>Recipient Beneficial Owner Originating Country Code</td>
<td>24</td>
<td>2</td>
<td>A</td>
<td>ISO 3166</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 Alpha or Blank</td>
</tr>
<tr>
<td>Recipient Beneficial Owner Originating Country Code</td>
<td>26</td>
<td>20</td>
<td>AN</td>
<td>Originating Country Tax</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Identification Number or Blank if data not available</td>
</tr>
<tr>
<td>Recipient OECD Recipient Type</td>
<td>46</td>
<td>2</td>
<td>AN</td>
<td>See Table 1.</td>
</tr>
<tr>
<td>Recipient Beneficial Owner Date of Birth</td>
<td>48</td>
<td>8</td>
<td>AN</td>
<td>ISO 8601: CCYYMMDD or CCYYMM or CCYY or Blank if unknown</td>
</tr>
<tr>
<td>Recipient Beneficial Owner Name Format Type</td>
<td>56</td>
<td>1</td>
<td>N</td>
<td>0 - Fixed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 - Free</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>If Free, all available name details will be given as a</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>left justified string in bytes 57 through 266.</td>
</tr>
<tr>
<td>Recipient Beneficial Owner Keyname</td>
<td>57</td>
<td>70</td>
<td>AN</td>
<td>Keyname is the family name for individuals and the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>business name for legal entities.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Other Names includes First</td>
</tr>
<tr>
<td>Recipient</td>
<td>127</td>
<td>70</td>
<td>AN</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Code</td>
<td>Length</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------</td>
<td>--------</td>
<td>------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Beneficial Owner Other Names</td>
<td>197</td>
<td>35</td>
<td>AN</td>
<td>Title includes: Mr, Mrs, Doctor, Sir, Professor etc.</td>
</tr>
<tr>
<td>Beneficial Owner Title</td>
<td>232</td>
<td>35</td>
<td>AN</td>
<td>Suffix includes: Esquire, Senior, Junior etc.</td>
</tr>
<tr>
<td>Beneficial Owner Gender</td>
<td>267</td>
<td>1</td>
<td>AN</td>
<td>F - Female; M - Male; N - Non Individual; U - Unknown</td>
</tr>
<tr>
<td>Birth City</td>
<td>268</td>
<td>35</td>
<td>AN</td>
<td></td>
</tr>
<tr>
<td>Birth City Sub Entity</td>
<td>303</td>
<td>35</td>
<td>AN</td>
<td></td>
</tr>
<tr>
<td>Birth Country Code</td>
<td>338</td>
<td>2</td>
<td>A</td>
<td>ISO 3166 2 Alpha or Blank</td>
</tr>
<tr>
<td>Alias or Other Name Format Type</td>
<td>340</td>
<td>1</td>
<td>N</td>
<td>0 - Fixed; 1 - Free; If Free, all available name details will be given as a left justified string in bytes 341 through 550</td>
</tr>
<tr>
<td>Keyname</td>
<td>341</td>
<td>70</td>
<td>AN</td>
<td>Keyname is the family name for individuals and the business name for legal entities.</td>
</tr>
<tr>
<td>Other Names</td>
<td>411</td>
<td>70</td>
<td>AN</td>
<td>Other Names includes First Name, Middle Name and/or initials.</td>
</tr>
<tr>
<td>Recipient</td>
<td>Beneficial Owner Alias or Other Title</td>
<td>481</td>
<td>35</td>
<td>AN</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------------------------</td>
<td>-----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>Recipient</td>
<td>Beneficial Owner Alias or Other Suffix</td>
<td>516</td>
<td>35</td>
<td>AN</td>
</tr>
<tr>
<td>Recipient</td>
<td>Beneficial Owner In Care of Name Format Type</td>
<td>551</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>Recipient</td>
<td>Beneficial Owner In Care Of Keyname</td>
<td>552</td>
<td>70</td>
<td>AN</td>
</tr>
<tr>
<td>Recipient</td>
<td>Beneficial Owner In Care of Other Names</td>
<td>622</td>
<td>70</td>
<td>AN</td>
</tr>
<tr>
<td>Recipient</td>
<td>Beneficial Owner In Care of Title</td>
<td>692</td>
<td>35</td>
<td>AN</td>
</tr>
<tr>
<td>Recipient</td>
<td>Beneficial Owner In Care of Suffix</td>
<td>727</td>
<td>35</td>
<td>AN</td>
</tr>
<tr>
<td>Recipient</td>
<td>Beneficial Owner Address Type</td>
<td>762</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>Recipient</td>
<td>Beneficial Owner Address Format Type</td>
<td>763</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>Column</td>
<td>Recipient\nBeneficial Owner\nAddress Street</td>
<td>764</td>
<td>70</td>
<td>AN</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------</td>
<td>-----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>Column</td>
<td>Recipient\nBeneficial Owner\nAddress City</td>
<td>834</td>
<td>35</td>
<td>AN</td>
</tr>
<tr>
<td>Column</td>
<td>Recipient\nBeneficial Owner\nAddress Country\nSub-entity</td>
<td>869</td>
<td>35</td>
<td>AN</td>
</tr>
<tr>
<td>Column</td>
<td>Recipient\nBeneficial Owner\nAddress Postal Code</td>
<td>904</td>
<td>9</td>
<td>AN</td>
</tr>
</tbody>
</table>
| Column | Recipient\nBeneficial Owner\nAddress Country\nCode | 913 | 2 | A | ISO 3166
|        |                                          |    |    | 2 Alpha or Blank |
| Column | Recipient\nBeneficial Owner\nOther Address Type | 915 | 1 | N | 0 - Residential or Business
|        |                                          |    |    | 1 - Registered Office
|        |                                          |    |    | 2 - Other or Unknown |
| Column | Recipient\nBeneficial Owner\nOther Address Format\nType | 916 | 1 | N | 0 - Fixed
|        |                                          |    |    | 1 - Free
<p>|        |                                          |    |    | If Free, all available address details will be given as a left justified string in bytes 917 through 1065 |
| Column | Recipient\nBeneficial Owner\nOther Address Street | 917 | 70 | AN |
| Column | Recipient\nBeneficial Owner\nOther Address City | 987 | 35 | AN |
| Column | Recipient\nBeneficial Owner\nOther Address Country\nSub-entity | 1022 | 35 | AN | State, Province etc. |</p>
<table>
<thead>
<tr>
<th>Recipient</th>
<th>1057</th>
<th>9</th>
<th>AN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beneficial Owner Other Address Postal Code</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recipient</th>
<th>1066</th>
<th>2</th>
<th>AN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beneficial Owner Other Address Country Code</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recipient</th>
<th>1068</th>
<th>2</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent or Intermediary Country Code for TIN 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recipient</th>
<th>1070</th>
<th>20</th>
<th>AN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent or Intermediary TIN 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recipient</th>
<th>1090</th>
<th>2</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent or Intermediary Country Code for TIN 2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recipient</th>
<th>1092</th>
<th>20</th>
<th>AN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent or Intermediary TIN 2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recipient</th>
<th>1112</th>
<th>1</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent or Intermediary Name Format Type</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recipient</th>
<th>1113</th>
<th>70</th>
<th>AN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent or Intermediary Keyname</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recipient</th>
<th>1183</th>
<th>70</th>
<th>AN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent or Intermediary Other Names</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recipient</th>
<th>1253</th>
<th>35</th>
<th>AN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent or Intermediary Title</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recipient</th>
<th>1288</th>
<th>35</th>
<th>AN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent or Intermediary Suffix</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Keyname is the family name for individuals and the business name for legal entities.

Other Names includes First Name, Middle Name and/or initials.

Title includes: Mr, Mrs, Doctor, Sir, Professor etc.

Suffix includes:
<table>
<thead>
<tr>
<th>Field</th>
<th>Start Byte</th>
<th>Length (Bytes)</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent or Intermediary Suffix</td>
<td>1323</td>
<td>1</td>
<td>N</td>
<td>0 - Fixed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 - Free</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>If Free, all available address details will be given as a left justified string in bytes 1324 through 1472</td>
</tr>
<tr>
<td>Address Format Type</td>
<td>1324</td>
<td>70</td>
<td>AN</td>
<td></td>
</tr>
<tr>
<td>Address Street</td>
<td>1394</td>
<td>35</td>
<td>AN</td>
<td></td>
</tr>
<tr>
<td>Address City</td>
<td>1394</td>
<td>35</td>
<td>AN</td>
<td>State, Province etc.</td>
</tr>
<tr>
<td>Address Sub-entity</td>
<td>1394</td>
<td>35</td>
<td>AN</td>
<td></td>
</tr>
<tr>
<td>Address Postal Code</td>
<td>1464</td>
<td>9</td>
<td>AN</td>
<td></td>
</tr>
<tr>
<td>Address Country Code</td>
<td>1473</td>
<td>2</td>
<td>AN</td>
<td>ISO 3166</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 Alpha or Blank</td>
</tr>
<tr>
<td>Actual Payer Country Code for TIN</td>
<td>1475</td>
<td>2</td>
<td>A</td>
<td>ISO 3166</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 Alpha or Blank</td>
</tr>
<tr>
<td></td>
<td>Actual Payer</td>
<td></td>
<td></td>
<td>Actual Payer</td>
</tr>
<tr>
<td>---</td>
<td>--------------</td>
<td>---</td>
<td>---</td>
<td>--------------</td>
</tr>
<tr>
<td></td>
<td>TIN 1</td>
<td>1477</td>
<td>20</td>
<td>AN</td>
</tr>
<tr>
<td></td>
<td>Country Code for TIN 2</td>
<td>1497</td>
<td>2</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>TIN 2</td>
<td>1499</td>
<td>20</td>
<td>AN</td>
</tr>
<tr>
<td></td>
<td>OECD Payer Code</td>
<td>1519</td>
<td>2</td>
<td>AN</td>
</tr>
<tr>
<td></td>
<td>Name Format Type</td>
<td>1521</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>Keyname</td>
<td>1522</td>
<td>70</td>
<td>AN</td>
</tr>
<tr>
<td></td>
<td>Other Names</td>
<td>1592</td>
<td>70</td>
<td>AN</td>
</tr>
<tr>
<td></td>
<td>Title</td>
<td>1662</td>
<td>35</td>
<td>AN</td>
</tr>
<tr>
<td></td>
<td>Suffix</td>
<td>1697</td>
<td>35</td>
<td>AN</td>
</tr>
<tr>
<td></td>
<td>Address Format Type</td>
<td>1732</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>Actual Payer Address Street</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Address Street</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>Actual Payer Address City</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Address City</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>Actual Payer Address Country Sub entity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Address Country Sub entity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>Actual Payer Address Postal Code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Address Postal Code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>Actual Payer Address Country Code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Address Country Code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>Payer Agent or Intermediary Country Code for TIN 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Payer Agent or Intermediary Country Code for TIN 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>Payer Agent or Intermediary TIN 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Payer Agent or Intermediary TIN 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>Payer Agent or Intermediary Country Code for TIN 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Payer Agent or Intermediary Country Code for TIN 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>Payer Agent or Intermediary TIN 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Payer Agent or Intermediary TIN 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>Payer Agent or Intermediary Name Format Type</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Payer Agent or Intermediary Name Format Type</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>77</td>
<td>Payer Agent or Intermediary Keyname</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Payer Agent or Intermediary Keyname</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>Payer Agent or Intermediary Other Names</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Payer Agent or Intermediary Other Names</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>79</td>
<td>Payer Other Names</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Keyname is the family name for individuals and the business name for legal entities.

Other Names includes First Name, Middle Name and/or initials.

Title includes:
Agent or Intermediary
Title

80  Payer
Agent or Intermediary
Suffix

2104  35  AN
Suffix includes:
Esquire, Senior, Junior etc.

81  Payer
Agent or Intermediary
Address Format Type

2139  1  N
0 - Fixed
1 - Free
If Free, all available name
details will be given as a left
justified string in bytes 2140
trough 2289

82  Payer
Agent or Intermediary
Address Street

2140  70  AN

83  Payer
Agent or Intermediary
Address City

2210  35  AN

84  Payer
Agent or Intermediary
Address Country Sub
entity

2245  35  AN
State, Province etc.

85  Payer
Agent or Intermediary
Address Postal Code

2380  9  AN

86  Payer
Agent or Intermediary
Address Country
Code

2289  2  A
ISO 3166
2 Alpha or Blank
<table>
<thead>
<tr>
<th>Field</th>
<th>Code</th>
<th>Length</th>
<th>Format</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sending Country</td>
<td>2291</td>
<td>8</td>
<td>AN</td>
<td>ISO 8601 CCYYMMDD or CCYYMM or CCYY or Blank if unknown</td>
</tr>
<tr>
<td>Tax Year End</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Date of Payment</td>
<td>2299</td>
<td>8</td>
<td>AN</td>
<td>ISO 8601 CCYYMMDD or CCYYMM or CCYY or Blank if unknown</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OECD Payment Type</td>
<td>2307</td>
<td>4</td>
<td>AN</td>
<td>See Table 2</td>
</tr>
<tr>
<td>Sending Country Payment Type</td>
<td>2311</td>
<td>4</td>
<td>AN</td>
<td></td>
</tr>
<tr>
<td>Gross Income Paid Currency Code</td>
<td>2315</td>
<td>3</td>
<td>A</td>
<td>ISO 4217 3 Alpha</td>
</tr>
<tr>
<td>Gross Income Paid Amount</td>
<td>2318</td>
<td>18</td>
<td>N</td>
<td>Major units of currency only</td>
</tr>
<tr>
<td>Major Unit of Currency</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net Income Paid Currency Code</td>
<td>2336</td>
<td>3</td>
<td>A</td>
<td>ISO 4217 3 Alpha</td>
</tr>
<tr>
<td>Net Income Paid Amount</td>
<td>2339</td>
<td>18</td>
<td>N</td>
<td>Major units of currency only</td>
</tr>
<tr>
<td>Major Unit of Currency</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tax Withheld Currency Code</td>
<td>2357</td>
<td>3</td>
<td>A</td>
<td>ISO 4217 3 Alpha</td>
</tr>
<tr>
<td>Tax Withheld Amount</td>
<td>2360</td>
<td>18</td>
<td>N</td>
<td>Major units of currency only</td>
</tr>
<tr>
<td>Major Unit of Currency</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tax Rate</td>
<td>2378</td>
<td>4</td>
<td>N</td>
<td>nn.nn decimal point virtual Blank if no data</td>
</tr>
<tr>
<td>Tax Refund Currency Code</td>
<td>2382</td>
<td>3</td>
<td>A</td>
<td>ISO 4217 3 Alpha</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Type</td>
<td>Length</td>
<td>Format</td>
</tr>
<tr>
<td>-------</td>
<td>------------------------------------------</td>
<td>------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>99</td>
<td>Tax Refund Amount Major Unit of Currency</td>
<td>2385</td>
<td>18</td>
<td>N</td>
</tr>
<tr>
<td>100</td>
<td>Date of Refund or Date of First Refund</td>
<td>2403</td>
<td>8</td>
<td>AN</td>
</tr>
<tr>
<td>101</td>
<td>Sender's Reference</td>
<td>2411</td>
<td>70</td>
<td>AN</td>
</tr>
<tr>
<td>102</td>
<td>Correction Reference</td>
<td>2481</td>
<td>70</td>
<td>AN</td>
</tr>
<tr>
<td>103</td>
<td>Filler General Use</td>
<td>2551</td>
<td>105</td>
<td>AN</td>
</tr>
<tr>
<td>104</td>
<td>Filler Specific Arrangements</td>
<td>2656</td>
<td>105</td>
<td>AN</td>
</tr>
</tbody>
</table>
Table 1: Type of recipient and type of payer

01 Individual
02 Corporation
03 Partnership
04 Business organisation other than corporation and partnership
05 Government or international organisation
06 Other (specify in filler)
07 Unknown

Table 2: Type of income code corresponding to the numbering of the Article of the OECD Model Convention on Income and Capital

6-- Income from immovable property
7-- 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 arrangements
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
2. **Standard for an OECD Foreign Data Exchange Magnetic Media Specifications**

**ORGANISATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT**

Foreign Tax Data Exchange
Magnetic Media Specifications

<table>
<thead>
<tr>
<th>Identity Information</th>
<th>Sending Country Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Receiving Country Name</td>
</tr>
<tr>
<td></td>
<td>Sending Country Contact Person’s Name/ Phone Number</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Media Information</th>
<th>Total Number of Media being sent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>File Name(s)</td>
</tr>
<tr>
<td></td>
<td>File Name Format: xxyzznn</td>
</tr>
<tr>
<td></td>
<td>xx= Sending Country</td>
</tr>
<tr>
<td></td>
<td>(Code ISO 3166 2 Alpha)</td>
</tr>
<tr>
<td></td>
<td>yy= Receiving Country</td>
</tr>
<tr>
<td></td>
<td>(Code ISO 3166 2 Alpha)</td>
</tr>
<tr>
<td></td>
<td>zz= Year of Sending</td>
</tr>
<tr>
<td></td>
<td>nn= Sequence Number of File</td>
</tr>
<tr>
<td></td>
<td>(nn-th File from xx to yy in zz)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Record Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Block Size</td>
</tr>
<tr>
<td></td>
<td>Density</td>
</tr>
<tr>
<td></td>
<td>Creation Date</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Tax Year(s) of Data</th>
</tr>
</thead>
</table>

Comments
____________________________________________________________________________
____________________________________________________________________________
____________________________________________________________________________

Signature of Competent Authority | Title | Date
3. **Standard for physical label attached to the magnetic media**

**ORGANISATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT**

**Foreign Tax Data Exchange**

**Physical Label Attached to Magnetic Media**

<table>
<thead>
<tr>
<th>Foreign Tax Data Exchanged in OECD Standard Magnetic Format</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sending Country Name:</strong></td>
</tr>
<tr>
<td><strong>Receiving Country Name:</strong></td>
</tr>
<tr>
<td><strong>Record Length:</strong></td>
</tr>
<tr>
<td><strong>Number of Records on Media:</strong></td>
</tr>
<tr>
<td><strong>Sequence Number of Media in this Transmission</strong> (n of m: n/m):</td>
</tr>
</tbody>
</table>

N.B.: The Tax Data provided are to be used according to the provisions of the international instrument under which they are exchanged.
The revised OECD Standard Magnetic Format (SMF 1997) is an improved version of the 1992 Standard Magnetic Format. The revised OECD Standard Magnetic Format includes the standards for (1) the Record Layout and two codetables, (2) OECD Foreign Data Exchange Magnetic Media Specifications - Transmittal Document -, (3) Physical label to be attached to the magnetic media.

**Record Layout Version 1997**

The international standards established by the International Organization for Standardization (ISO codes) are used as codes for country names, currencies, dates, definition of characters used, file structure and labelling of the media. Two tables indicate the code for type of recipient and type of payer as well as the code for income. The record layout includes 104 fields, including filler fields at the end: fields 2 to 40 are allocated to the recipient beneficial owner, fields 41 to 55 are allocated to the recipient beneficial owner's agent or intermediary, fields 56 to 71 are allocated to the actual payer of the income, fields 72 to 86 are allocated to the payer's agent or intermediary. For each series of fields the same pattern is followed to provide information on the Tax Identification Number (TIN), name, alias or other name and address. Fields 87 to 100 are allocated to the income (tax year, date, type of payment, currency, gross and net amount, tax withheld, refund etc.). Fields 101 and 102 are allocated to record references. Fields 103 and 104 are allocated to fillers to provide additional information. For each field are provided: (1) the number of the field, (2) the starting position of the field, (3) its length, (4) the data type: A=alphabetic, N= numeric, AN= alphanumeric, B= blank. Entries to all fields marked as alphabetic or alphanumeric are to be left-justified. Entries to fields marked as numeric are to be right-justified. If the content of a field does not fill the entire space provided for this field, the rest of the field shall be filled with blanks; in the case of a numeric field it is also acceptable to fill with leading zeros.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F001</td>
<td>***Document Type Indicator</td>
<td>1</td>
<td>1</td>
<td>n</td>
<td>&quot;0&quot; repeated &quot;1&quot; new &quot;2&quot; correction</td>
</tr>
</tbody>
</table>

This field is used to indicate whether the following data are new, a replication of data already sent (possibly not received) or corrections to data sent. The field applies only to the one record in which it is included. In the case of repeated or corrected data field #102 must contain the identifier (field #101) of the record it refers to. Whenever the reference to a replicated record is not found by the receiving country it will be assumed to be
new data. If the reference to a correcting record is not found, the correction is treated as replication. Records shall be transmitted and, what is even more important, processed in the following order: repeated - new - correction. In the case of a correction also the unchanged fields shall be transmitted again (i.e., repeated) - except for field #101.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBO</td>
<td>***Recipient Beneficial Owner</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fields #2 to #40 contain data about the Recipient Beneficial Owner. 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.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F002</td>
<td>**ResidenceCountryCode</td>
<td>2</td>
<td>2</td>
<td>a</td>
<td>ISO 3166 ; blank if unknown</td>
</tr>
</tbody>
</table>

Enter the appropriate country code for the state of which the recipient beneficial owner is a resident from the ISO 3166 two-byte alpha version. Leave blank if no data are available. Most likely, however, the "Unknown" case will not happen in a transmission as you would not know where to transmit the data.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F003</td>
<td>**ResidenceCountryTIN</td>
<td>4</td>
<td>20</td>
<td>an</td>
<td>Residence Country Tax Identification Number or blank if data not available</td>
</tr>
</tbody>
</table>

Enter the Tax Identification Number (TIN) of the recipient beneficial owner in the state of which he is a resident for treaty purposes. Leave blank if no data are available. If in the residence country a TIN properly said does not exist but another identifier is used regularly in that country for tax purposes it can be entered here instead of a TIN. Other (general) identifiers can be helpful for matching; they shall, however, be entered in field #104 together with an explanation of their type. The country specific information of the SMF documentation or the technical note accompanying the transmission also may contain further advice.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F004</td>
<td>**OriginatingCountryCode</td>
<td>24</td>
<td>2</td>
<td>a</td>
<td>ISO 3166 ; blank if unknown</td>
</tr>
<tr>
<td>Field/Field_Group</td>
<td>Field/Group_Name</td>
<td>Start</td>
<td>Length</td>
<td>Data_Type</td>
<td>Remarks</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------</td>
<td>-------</td>
<td>--------</td>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>F005</td>
<td><strong>OriginatingCountryTIN</strong></td>
<td>26</td>
<td>20</td>
<td>an</td>
<td>Enter the Tax Identification Number (TIN) of the recipient beneficial owner assigned by the state out of which the income is paid. This instruction applies in the case of countries - like the United States - that also assign TINs to non-residents. The type of such number might be explained in field #104. The country-specific explanation of the SMF documentation also may contain further information which may help the treaty partner decide, whether it can make use of such &quot;TIN&quot;. Leave blank if no data are available.</td>
</tr>
<tr>
<td>F006</td>
<td><strong>OECDRecipientType</strong></td>
<td>46</td>
<td>2</td>
<td>an</td>
<td>Enter the appropriate type of the recipient from codelist #1.</td>
</tr>
<tr>
<td>F007</td>
<td><strong>DateOfBirth</strong></td>
<td>48</td>
<td>8</td>
<td>an</td>
<td>Enter the date of birth of the recipient beneficial owner as indicated in the ISO 8601 standard, i.e., CCYYMMDD; or CCYYMM; or CCYY (C = Century; Y = Year; M = Month; D = Day). Leave blank if no data are available.</td>
</tr>
<tr>
<td>N_RBO</td>
<td><strong>(name)</strong></td>
<td></td>
<td></td>
<td></td>
<td>The name of the recipient beneficial owner as indicated to the sending country by payer documentation or official records. In the fields #8 to #12, the user must enter data about the name of the recipient beneficial owner. Refer to those fields for more information. If the recipient beneficial owner is also known under another name (&quot;alias&quot;), insert that data in fields #17 to #21. Refer to those fields for more information.</td>
</tr>
<tr>
<td>F008</td>
<td>*NameFormatType</td>
<td>56</td>
<td>1</td>
<td>n</td>
<td>For fields #9 to #12, the user has the option to enter the data about the name of the recipient beneficial owner either as one long left-justified field of 210 bytes in positions 57 through 266, or to spread the data over 4 fields. If the user chooses the option to enter the data required in fields #9 to</td>
</tr>
</tbody>
</table>
#12 in one field, enter "1". This will be the common case with legal entities. All available name details should then be entered as one left-justified string of 210 bytes, blanks serving as delimiter between name parts. If the user chooses the option to enter the data required in fields #9 to #12 in separate fields, enter "0" (zero). The free format should be used if there is any doubt about the name details (which is keyname, which is first name etc.) as the receiving country may be in a better position to make a correct judgment. Note that "keyname" is intended to be a culturally neutral notion for what otherwise might be called "surname", "last name", "family name" etc.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>_<em><strong><strong><strong><strong><strong><strong><strong>Field/Group_Name</strong></strong></strong></strong></strong></strong></strong></em></th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreeForm1</td>
<td>*(name free format)</td>
<td>57</td>
<td>210</td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

If the user has entered "1" in field #8 he has opted to enter the data about the name of the recipient beneficial owner in one long left-justified field of 210 bytes. For individuals enter the family name, the first name, middle name or initials (if available), the title (if available, e.g. Mr., Mrs.), and/or the suffix (if available, e.g., Esquire, Senior). For legal entities (corporations, partnerships, etc.), enter the registered or documented business name. Use blank as a delimiter between parts of the name.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>_<em><strong><strong><strong><strong><strong><strong><strong>Field/Group_Name</strong></strong></strong></strong></strong></strong></strong></em></th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>FixForm1</td>
<td>*(name fixed format)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #8, he has opted to enter the data about the name of the recipient beneficial owner in four separate fields. In fields #9 to #12, enter for individuals the family name (#9), the first name, middle name and/or initials (if available) (#10), the title (if available, e.g. Mr., Mrs.) (#11) and/or the suffix (if available, e.g. Esquire, Senior) (#12). Note that "keyname" is intended to be a culturally neutral notion for what otherwise might be called "surname", "last name", "family name" etc. For legal entities (corporations, partnerships, etc.) if you do not use free form as recommended, enter the complete business name in field #9.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>_<em><strong><strong><strong><strong><strong><strong><strong>Field/Group_Name</strong></strong></strong></strong></strong></strong></strong></em></th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F009</td>
<td>Keyname</td>
<td>57</td>
<td>70</td>
<td>an</td>
<td>Keyname is the family name for individuals and the business name for legal entities</td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #8, he has opted to enter the data about the name of the recipient beneficial owner in four separate fields. In field #9 enter the family name for individuals or the business name for legal entities (if you do not use free form as recommended for legal entities).

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>_<em><strong><strong><strong><strong><strong><strong><strong>Field/Group_Name</strong></strong></strong></strong></strong></strong></strong></em></th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F010</td>
<td>OtherNames</td>
<td>127</td>
<td>70</td>
<td>an</td>
<td>Other name includes first name, middle name and/or initials</td>
</tr>
</tbody>
</table>
If the user has entered "0" (zero) in field #8, he has opted to enter the data about the name of the recipient beneficial owner in four separate fields. In field #10, enter the first name, the middle name and/or the initials. Leave blank if no data are available or in the case of a legal entity.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F011</td>
<td>Title</td>
<td>197</td>
<td>35</td>
<td>an</td>
<td>Title includes: Mr, Mrs, Doctor, Sir, Professor etc.</td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #8, he has opted to enter the data about the name of the recipient beneficial owner in four separate fields. In field #11, enter the title. Titles may include Mr; Mrs; Doctor, Sir, Professor, etc. Abbreviations may be used as customary. Leave blank if no data are available or in the case of a legal entity.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F012</td>
<td>Suffix</td>
<td>232</td>
<td>35</td>
<td>an</td>
<td>Suffix includes: Esquire, Senior, Junior etc.</td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #8, he has opted to enter the data about the name of the recipient beneficial owner in four separate fields. In field #12, enter the suffix. Suffixes may include Esquire, Senior, Junior, etc. Abbreviations may be used as customary. Leave blank if no data are available or in the case of a legal entity.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F013</td>
<td><strong>Gender</strong></td>
<td>267</td>
<td>1</td>
<td>a</td>
<td>F - Female; M - Male; N - Non individual; U - Unknown</td>
</tr>
</tbody>
</table>

The Recipient Beneficial Owner of the income may be an individual, a company, a partnership, or other business organisation or a permanent establishment. For all cases except individuals enter "N". Note that this includes partnerships. Example: a partnership of two male partners will be denoted "N". Individuals are identified as "F" or "M" or "U" as the case may be. The transmitting country shall use upper case; however, the country receiving the information shall be able to process lower case as well as upper case.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F014</td>
<td><strong>BirthCity</strong></td>
<td>268</td>
<td>35</td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

Enter the name of the city where the recipient beneficial owner was born. This field may serve to better identify the beneficial owner. It applies only for natural persons. This field may serve to better identify the beneficial owner. It applies only for natural persons. Leave blank if no data are available or in the case of a legal entity.
Enter the name of the sub-entity of the city where the recipient beneficial owner was born. This field may serve to better identify the beneficial owner. It applies only for natural persons. Leave blank if no data are available or in the case of a legal entity. This field will generally be used only in the case of large cities that are subdivided, e.g., Manhattan (New York), Kreuzberg (Berlin).

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F015 <strong>BirthCitySubentity</strong></td>
<td></td>
<td>303</td>
<td>35</td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

Enter the appropriate country code for the state in which the city of birth of the recipient beneficial owner is situated from the ISO 3166 two-byte alpha version. Leave blank if no data are available or in the case of a legal entity.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F016 <strong>BirthCountryCode</strong></td>
<td></td>
<td>338</td>
<td>2</td>
<td>a</td>
<td>ISO 3166 2 Alpha or Blank</td>
</tr>
</tbody>
</table>

Sometimes the recipient beneficial owner is also known under another name (an "alias"). For legal entities (corporations, partnerships etc.) an alias might be a short name of the entity or a name that is used for public acquaintance instead of the official business name ("doing business as -DBA", "trading as - TA"); examples: in the United States, DaimlerChrysler is still known simply as Chrysler, Dr. William Black DAB Quality Pediatrics, Inc.). For such alias names enter available data in fields #17 to #21. Leave blank if no data are available (which will probably be the most common case). Instructions for these fields will more or less duplicate those for fields #8 to #12. However, fields that are numeric in the "name"-group are defined alphanumeric here to allow for a blank value when no Alias is known.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALI <strong>(AliasOrOther)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For fields #18 through #21, the user has the option to enter the data about the "alias" name of the recipient beneficial owner either as one long left-justified field of 210 bytes in positions 341 through 550, or to spread the data over 4 fields. If the user chooses the option to enter the data required in fields #18 through #21 in one field, enter "1". This will be the common case with legal entities. All available name details should then be entered as one left-justified string of 210 bytes, blanks serving as delimiter between name parts. If the user chooses the option to enter the data required in fields #18 through #21 in separate fields, enter "0" (zero). The free format should be used if there is any doubt about the name details (which is keyname, which is first name etc.) as the receiving country may be in a better position to make a correct judgment. Note that "keyname" is intended to be a culturally neutral notion for what otherwise might be called "surname", "last name", "family name" etc. To allow for the whole "alias" group
of fields to be left blank if no such information is known, this field is defined alphanumeric in contrast to the "name" group of fields and a blank is a valid entry in the case mentioned.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreeForm2</td>
<td><em>(name free format)</em></td>
<td>341</td>
<td>210</td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

If the user has entered "1" in field #17 he has opted to enter the data about the alias of the recipient beneficial owner in one long left-justified field of 210 bytes. For individuals enter the family name, the first name, middle name or initials (if available), the title (if available, e.g. Mr., Mrs.), and/or the suffix (if available, e.g., Esquire, Senior). For legal entities (corporations, partnerships, etc.), enter the registered or documented business name. Use blank as a delimiter between parts of the name.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>FixForm2</td>
<td><em>(name fixed format)</em></td>
<td></td>
<td></td>
<td></td>
<td>The details of the subdivision in fields are given below.</td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #17, he has opted to enter the data about the alias of the recipient beneficial owner in four separate fields. In fields #18 to #21, enter for individuals the family name (#18), the first name, middle name and/or initials (if available) (#19), the title (if available, e.g. Mr., Mrs.) (#20) and/or the suffix (if available, e.g. Esquire, Senior) (#21). Note that "keyname" is intended to be a culturally neutral notion for what otherwise might be called "surname", "last name", "family name" etc. For legal entities (corporations, partnerships, etc.) if you do not use free form as recommended, enter the complete business name in field #18.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F018</td>
<td>Keyname</td>
<td>341</td>
<td>70</td>
<td>an</td>
<td>Keyname is the family name for individuals and the business name for legal entities</td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #17, he has opted to enter the data about the alias of the recipient beneficial owner in four separate fields. In field #18 enter the family name for individuals or the business name for legal entities (if you do not use free form as recommended for legal entities).

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F019</td>
<td>OtherNames</td>
<td>411</td>
<td>70</td>
<td>an</td>
<td>Other name includes first name, middle name and/or initials</td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #17, he has opted to enter the data about the alias of the recipient beneficial owner in four separate fields. In field #19, enter the first name, the middle name and/or the initials. Leave blank if no data are available or in the case of a legal entity.
If the user has entered "0" (zero) in field #17, he has opted to enter the data about the alias of the recipient beneficial owner in four separate fields. In field #20, enter the title. Titles may include Mr; Mrs; Doctor, Sir; Professor, etc. Abbreviations may be used as customary. Leave blank if no data are available or in the case of a legal entity.

If the user has entered "0" (zero) in field #17, he has opted to enter the data about the alias of the recipient beneficial owner in four separate fields. In field #21, enter the suffix. Suffixes may include Esquire, Senior, Junior, etc. Abbreviations may be used as customary. Leave blank if no data are available or in the case of a legal entity.

A person that is acting for the recipient beneficial owner in some capacity (e.g., legal, administrative, or functional). This can be something as simple as the use of another person's post office box for the receipt of mail and does not necessarily imply the creation of any legal obligation or responsibility.

For fields #23 to #26, the user has the option to enter the data about the name of the in-care-of person either as one long left-justified field of 210 bytes in positions 552 through 761, or to spread the data over 4 fields. If the user chooses the option to enter the data required in fields #23 to #26 in one field, enter "1". This will be the common case with legal entities. All available name details should then be entered as one left-justified string of 210 bytes, blanks serving as delimiter between name parts. If the user chooses the option to enter the data required in fields #23 to #26 in separate fields, enter "0" (zero). The free format should be used if there is any doubt about the name details (which is keyname, which is first name etc.) as the receiving country may be in a better position to make a correct judgment. Note that "keyname" is intended to be a culturally neutral notion for what otherwise might be called "surname", "last name", "family name", etc. To allow for the whole "in-care-of" group of fields to be left blank if no
such information is known, this field is defined alphanumeric in contrast to the "name" group of fields and a blank is a valid entry in the case mentioned.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreeForm3</td>
<td>*(name free format)</td>
<td>552</td>
<td>210</td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

If the user has entered "1" in field #22 he has opted to enter the data about the in-care-of person in one long left-justified field of 210 bytes. For individuals enter the family name, the first name, middle name or initials (if available), the title (if available, e.g. Mr., Mrs.), and/or the suffix (if available, e.g., Esquire, Senior). For legal entities (corporations, partnerships, etc.), enter the registered or documented business name. Use blank as a delimiter between parts of the name.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>FixForm3</td>
<td>*(name fixed format)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The details of the subdivision in fields are given below.

If the user has entered "0" (zero) in field #22, he has opted to enter the data about the in-care-of person in four separate fields. In fields #23 to #26, enter for individuals the family name (#23), the first name, middle name and/or initials (if available) (#24), the title (if available, e.g. Mr., Mrs.) (#25) and/or the suffix (if available, e.g. Esquire, Senior) (#26). Note that "keyname" is intended to be a culturally neutral notion for what otherwise might be called "surname", "last name", "family name" etc. For legal entities (corporations, partnerships, etc.) if you do not use free form as recommended, enter the complete business name in field #23.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F023</td>
<td>Keyname</td>
<td>552</td>
<td>70</td>
<td>an</td>
<td>Keyname is the family name for individuals and the business name for legal entities</td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #22, he has opted to enter the data about the in-care-of person in four separate fields. In field #23 enter the family name for individuals or the business name for legal entities (if you do not use free form as recommended for legal entities).

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F024</td>
<td>OtherNames</td>
<td>622</td>
<td>70</td>
<td>an</td>
<td>Other name includes first name, middle name and/or initials</td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #22, he has opted to enter the data about the in-care-of person in four separate fields. In field #24, enter the first name, the middle name and/or the initials. Leave blank if no data are available or in the case of a legal entity.
<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F025</td>
<td>Title</td>
<td>692</td>
<td>35</td>
<td>an</td>
<td>Title includes: Mr, Mrs, Doctor, Sir, Professor etc.</td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #22, he has opted to enter the data about the in-care-of person in four separate fields. In field #25, enter the title. Titles may include Mr; Mrs; Doctor, Sir, Professor, etc. Abbreviations may be used as customary. Leave blank if no data are available or in the case of a legal entity.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F026</td>
<td>Suffix</td>
<td>727</td>
<td>35</td>
<td>an</td>
<td>Suffix includes: Esquire, Senior, Junior etc.</td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #22, he has opted to enter the data about the in-care-of person in four separate fields. In field #26, enter the suffix. Suffixes may include Esquire, Senior, Junior, etc. Abbreviations may be used as customary. Leave blank if no data are available or in the case of a legal entity.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>A_RBO</td>
<td>**(Address)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fields #27 to #33 contain the primary or usual address of the recipient beneficial owner as documented to the paying entity or sending administration. The main aim of including this address is for identification of the recipient beneficial owner and to give the means of sending mail. The location of this address does not necessarily imply any tax relevance. The address can basically be situated in any country. The sending country may also inform the receiving country of another address it has identified in the next field group (fields #34 through #40).

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F027</td>
<td>*AddressType</td>
<td>762</td>
<td>1</td>
<td>n</td>
<td>0 - Residential or Business; 1 - Registered Office; 2 - Other or Unknown</td>
</tr>
</tbody>
</table>

Enter the appropriate type of the address of the recipient beneficial owner by selecting from the given possibilities: 0 - Residential or Business; 1 - Registered Office; 2 - Other or Unknown. Due to the fact that the address usually relates to a resident of another state, the sending state may not be certain of the type of the address. Therefore, when in doubt, the user should enter "2".
For fields #29 through #32, the user has the option to enter the data about the address of the recipient beneficial owner either as one long field or to spread the data over four fields. If the user chooses the option to enter the data required in fields #29 through #32 in separate fields, enter "0" (zero). If the user chooses the option to enter the data required in a less structured way enter "1". All available details on the address of the recipient beneficial owner should then be presented as one left-justified string of 149 bytes, blank or "/" (slash) used as a delimiter between parts of the address. PLEASE NOTE that the address country code is outside this area of choosable format. 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.

If the user has entered "1" in field #28, he has opted to enter the data about the address of the recipient beneficial owner in one long left-justified field of 149 bytes. Enter the street and number, the city, the country sub-entity - if applicable (state, province or other political subunit) - and the postal (zip) code. Use blank as a delimiter between parts of the address. You may also use "/" (slash) as a delimiter to designate a new line in the printed form of the address. Do not enter the address country code in this field. It is strongly recommended that the address information be presented in the way the state where the recipient beneficial owner resides or is located usually structures the address. Refer to the structure information to which the link "ad-strct" points - depending on implementation.

If the user has entered "0" (zero) in field #28, he has opted to enter the data about the address of the recipient beneficial owner in four separate fields (#29 through #32).

If the user has entered "0" (zero) in field #28, he has opted to enter the data about the address of the recipient beneficial owner in four separate fields (#29 through #32). In field #29, enter the complete name of the street and the building number where the recipient beneficial owner resides or
is situated. It is strongly recommended that the address information be presented in the way the state where the recipient beneficial owner resides or is located usually structures the address. Refer to the structure information to which the link "ad-strct" points - depending on implementation.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F030</td>
<td>City</td>
<td>834</td>
<td>35</td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #28, he has opted to enter the data about the address of the recipient beneficial owner in four separate fields (#29 through #32). In field #30, enter the name of the city or municipality where the recipient beneficial owner resides or is situated.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F031</td>
<td>CountrySubentity</td>
<td>869</td>
<td>35</td>
<td>an</td>
<td>State, Province etc</td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #28, he has opted to enter the data about the address of the recipient beneficial owner in four separate fields (#29 through #32). In field #31, enter the name of the country sub-entity if applicable in which the city where the recipient beneficial owner resides or is situated. Sub-entities may include states, regions, laender, provinces, or other political subunits. Leave blank if no data are available.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F032</td>
<td>PostalCode</td>
<td>904</td>
<td>9</td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #28, he has opted to enter the data about the address of the recipient beneficial owner in four separate fields (#29 through #32). In field #32, enter the postal (zip) code of the address of the recipient beneficial owner. It is strongly recommended that the address information be presented in the way the state where the recipient beneficial owner resides or is located usually structures the address. Refer to the structure information to which the link "ad-strct" points - depending on implementation.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F033</td>
<td>*CountryCode</td>
<td>913</td>
<td>2</td>
<td>a</td>
<td>ISO 3166 ; blank if unknown</td>
</tr>
</tbody>
</table>

Enter the ISO 3166 two-byte alpha country code of the address entered in bytes 764 through 912. This field is not included in the alternatively fixed or free area in order to provide a unique location for this information.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>OA_RBO</td>
<td>**(OtherAddress)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This field group (fields #35 through #39) offers the opportunity to inform the information receiving country of a second address for the beneficial owner of which the information sending country has knowledge. The structure of this field group is identical to the primary address field group.
This field group may be left empty (blank), if no other address is known. Therefore some fields that are numeric only in the primary address field group are defined alphanumeric here to allow for a blank value.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F034</td>
<td>*OtherAddressType</td>
<td>915</td>
<td>1</td>
<td>an</td>
<td>0 - Residential or Business; 1 - Registered Office; 2 - Other or Unknown; blank is allowed if the whole field group is empty.</td>
</tr>
</tbody>
</table>

Enter the appropriate type of the address of the recipient beneficial owner by selecting from the given possibilities: 0 - Residential or Business; 1 - Registered Office; 2 - Other or Unknown. Due to the fact that the address usually relates to a resident of another state, the sending state may not be certain of the type of the address. Therefore, when in doubt, the user should enter "2".

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F035</td>
<td>*OtherAddrFormatType</td>
<td>916</td>
<td>1</td>
<td>an</td>
<td>0 fixed; 1 free; blank if no other address given. If free, all available address details will be given as a left justified string in bytes 917 through 1065.</td>
</tr>
</tbody>
</table>

For fields #36 through #39, the user has the option to enter the data about the other address of the recipient beneficial owner either as one long field or to spread the data over four fields. If the user chooses the option to enter the data required in fields #36 through #39 in separate fields, enter "0" (zero). If the user chooses the option to enter the data required in a less structured way enter "1". All available details on the other address of the recipient beneficial owner should then be presented as one left-justified string of 149 bytes, blank or "/" (slash) used as a delimiter between parts of the address. PLEASE NOTE that the address country code is outside this area of choosable format. 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.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreeForm5</td>
<td>*(other address free format)</td>
<td>917</td>
<td>149</td>
<td>an</td>
<td>If the user has entered &quot;1&quot; in field #35, he has opted to enter the data about the other address of the recipient beneficial owner in one long left-justified field of 149 bytes. Enter the street and number, the city, the country sub-entity - if applicable (state, province or other political subunit) - and the postal (zip) code. Use blank as a delimiter between parts of the address. You may also use &quot;/&quot; (slash) as a delimiter to designate a new line in the printed form of the address. Do not enter the address country code in this field. It is strongly recommended that the address information be presented in the way the state where the recipient beneficial owner resides or is located usually structures the address. Refer to the structure information to which the link &quot;ad-strct&quot; points - depending on implementation.</td>
</tr>
</tbody>
</table>

35
The details of the subdivision in fields are given below.

If the user has entered "0" (zero) in field #35, he has opted to enter the data about the other address of the recipient beneficial owner in four separate fields (#36 through #39).

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>FixForm5</td>
<td><em>(other address fixed format)</em></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #35, he has opted to enter the data about the other address of the recipient beneficial owner in four separate fields (#36 through #39). In field #36, enter the complete name of the street and the building number where the recipient beneficial owner resides or is situated. It is strongly recommended that the address information be presented in the way the state where the recipient beneficial owner resides or is located usually structures the address. Refer to the structure information to which the link "ad-strct" points - depending on implementation.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F036</td>
<td>Street</td>
<td>917 70</td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #35, he has opted to enter the data about the other address of the recipient beneficial owner in four separate fields (#36 through #39). In field #37, enter the name of the city or municipality where the recipient beneficial owner resides or is situated.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F037</td>
<td>City</td>
<td>987 35</td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #35, he has opted to enter the data about the other address of the recipient beneficial owner in four separate fields (#36 through #39). In field #38, enter the name of the country sub-entity if applicable in which the city where the recipient beneficial owner resides or is situated. Sub-entities may include states, regions, laender, provinces, or other political subunits. Leave blank if no data are available.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F038</td>
<td>CountrySubentity</td>
<td>1022 35</td>
<td>an</td>
<td>State, Province etc</td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #35, he has opted to enter the data about the other address of the recipient beneficial owner in four separate fields (#36 through #39). In field #39, enter the postal (zip) code of the other address of the recipient beneficial owner. It is strongly recommended that the address information be presented in the way the state where the recipient beneficial owner resides or is located usually structures the address. Refer to the structure information to which the link "ad-strct" points - depending on implementation.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F039</td>
<td>PostalCode</td>
<td>1057 9</td>
<td>an</td>
<td></td>
</tr>
<tr>
<td>Field/Field_Group</td>
<td>Field/Group_Name</td>
<td>Start</td>
<td>Length</td>
<td>Data_Type</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------</td>
<td>-------</td>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>F040</td>
<td>*CountryCode</td>
<td>1066</td>
<td>2</td>
<td>a</td>
</tr>
</tbody>
</table>

Enter the ISO 3166 two-byte alpha country code of the address entered in bytes 917 through 1065. This field is not included in the alternatively fixed or free area in order to provide a unique location for this information.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAI</td>
<td>***RecipientAgent Or Intermediary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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). If no such party is identified the field group should be left blank - this may even apply to those fields which would have to be numeric in the "existence" case.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F041</td>
<td>**CountryCodeForTIN1</td>
<td>1068</td>
<td>2</td>
<td>a</td>
<td>ISO 3166 ; blank if unknown</td>
</tr>
</tbody>
</table>

If a TIN is entered in field #42, enter the ISO 3166 two-byte alpha country code of the state that issued the TIN. Note that the agent or intermediary may reside or do business anywhere, therefore this can be any country.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F042</td>
<td>**TIN1</td>
<td>1070</td>
<td>20</td>
<td>an</td>
<td>Agent or Intermediary Tax Identification Number or Blank if data not available</td>
</tr>
</tbody>
</table>

If applicable, enter the TIN that the country identified in country code field #41 has issued to the agent or intermediary.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F043</td>
<td>**CountryCodeForTIN2</td>
<td>1090</td>
<td>2</td>
<td>a</td>
<td>ISO 3166 ; blank if unknown</td>
</tr>
</tbody>
</table>

If a TIN is entered in field #44, enter the ISO 3166 two-byte alpha country code of the state that issued the TIN. Note that the agent or intermediary may reside or do business anywhere, therefore this can be any country.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F044</td>
<td>**TIN2</td>
<td>1092</td>
<td>20</td>
<td>an</td>
<td>Agent or Intermediary Tax Identification Number or Blank if data not available</td>
</tr>
</tbody>
</table>
If applicable, enter the TIN that the country identified in country code field #43 has issued to the agent or intermediary. There is no rule whatsoever, if TINs assigned by two countries to the same recipient agent or intermediary are known, which TIN to enter as TIN1 and which as TIN2. However, the two numbers should not be identical (i.e., do not enter the same number both in fields #42 and #44).

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>N_RAI</td>
<td><strong>(name)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This is the name of the beneficial owner's agent or intermediary as indicated to the sending country by appropriate documentation or official records. In fields #45 through #49, the user should enter data about the name of the recipient beneficial owner's agent or intermediary if applicable. Refer to those fields for more information.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F045</td>
<td>*NameFormatType</td>
<td>1112</td>
<td>1</td>
<td>an</td>
<td>0 fixed; 1 free; blank acceptable if no agent or intermediary given. If free, all available name details will be given as a left justified string in bytes 1113 through 1322</td>
</tr>
</tbody>
</table>

For fields #46 through #49, the user has the option to enter the data about the name of the recipient beneficial owner's agent or intermediary either as one long left-justified field of 210 bytes in positions 1113 through 1322, or to spread the data over 4 fields. If the user chooses the option to enter the data required in fields #46 through #49 in one field, enter "1". This will be the common case with legal entities. All available name details should then be entered as one left-justified string of 210 bytes, blanks serving as delimiter between name parts. If the user chooses the option to enter the data required in fields #46 through #49 in separate fields, enter "0" (zero). The free format should be used if there is any doubt about the name details (which is keyname, which is first name etc.) as the receiving country may be in a better position to make a correct judgment. Note that "keyname" is intended to be a culturally neutral notion for what otherwise might be called "surname", "last name", "family name" etc. To allow for the whole "agent or intermediary" group of fields to be left blank if no such information is known, this field is defined alphanumeric in contrast to the "name" group of fields and a blank is a valid entry in the case mentioned.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreeForm6</td>
<td>*(name free format)</td>
<td>1113</td>
<td>210</td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

If the user has entered "1" in field #45 he has opted to enter the data about the name of the recipient beneficial owner's agent or intermediary in one long left-justified field of 210 bytes. For individuals enter the family name, the first name, middle name or initials (if available), the title (if available, e.g. Mr., Mrs.), and/or the suffix (if available, e.g., Esquire, Senior). For legal entities (corporations, partnerships, etc.), enter the registered or documented business name. Use blank as a delimiter between parts of the name. Leave blank if no data are available.
The details of the subdivision in fields are given below.

If the user has entered "0" (zero) in field #45, he has opted to enter the data about the name of the recipient beneficial owner’s agent or intermediary in four separate fields. In fields #46 to #49, enter for individuals the family name (#46), the first name, middle name and/or initials (if available) (#47), the title (if available, e.g. Mr., Mrs.) (#48) and/or the suffix (if available, e.g. Esquire, Senior) (#49). Note that "keyname" is intended to be a culturally neutral notion for what otherwise might be called "surname", "last name", "family name" etc. For legal entities (corporations, partnerships, etc.) if you do not use free form as recommended, enter the complete business name in field #46.

<table>
<thead>
<tr>
<th>Field/Field Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>FixForm6</td>
<td><em>(name fixed format)</em></td>
<td></td>
<td></td>
<td></td>
<td>The details of the subdivision in fields are given below.</td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #45, he has opted to enter the data about the name of the recipient beneficial owner’s agent or intermediary in four separate fields. In field #46 enter the family name for individuals or the business name for legal entities (if you do not use free form as recommended for legal entities).

<table>
<thead>
<tr>
<th>Field/Field Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F046</td>
<td>Keyname</td>
<td>1113</td>
<td>70</td>
<td>an</td>
<td>Keyname is the family name for individuals and the business name for legal entities</td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #45, he has opted to enter the data about the name of the recipient beneficial owner’s agent or intermediary in four separate fields. In field #47, enter the first name, the middle name and/or the initials. Leave blank if no data are available or in the case of a legal entity.

<table>
<thead>
<tr>
<th>Field/Field Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F047</td>
<td>OtherNames</td>
<td>1183</td>
<td>70</td>
<td>an</td>
<td>Other name includes first name, middle name and/or initials</td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #45, he has opted to enter the data about the name of the recipient beneficial owner’s agent or intermediary in four separate fields. In field #48, enter the title. Titles may include Mr; Mrs; Doctor, Sir, Professor, etc. Abbreviations may be used as customary. Leave blank if no data are available or in the case of a legal entity.

<table>
<thead>
<tr>
<th>Field/Field Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F048</td>
<td>Title</td>
<td>1253</td>
<td>35</td>
<td>an</td>
<td>Title includes: Mr, Mrs, Doctor, Sir, Professor etc.</td>
</tr>
</tbody>
</table>

39
**CTPA/CFA(2006)25/ADD3**

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F049</td>
<td>Suffix</td>
<td>1288</td>
<td>35</td>
<td>an</td>
<td>Suffix includes: Esquire, Senior, Junior etc.</td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #45, he has opted to enter the data about the name of the recipient beneficial owner's agent or intermediary in four separate fields. In field #49, enter the suffix. Suffixes may include Esquire, Senior, Junior, etc. Abbreviations may be used as customary. Leave blank if no data are available or in the case of a legal entity.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>A_RAI</td>
<td><strong>(Address)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This group of fields is used to enter address information about the recipient beneficial owner's agent or intermediary.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F050</td>
<td>*AddrFormatType</td>
<td>1323</td>
<td>1</td>
<td>an</td>
<td>0 fixed; 1 free; blank acceptable if no agent or intermediary given. If free, all available address details will be given as a left justified string in bytes 1324 through 1472.</td>
</tr>
</tbody>
</table>

For fields #51 through #54, the user has the option to enter the data about the address of the recipient beneficial owner's agent or intermediary either as one long field or to spread the data over four fields. If the user chooses the option to enter the data required in fields #51 through #54 in separate fields, enter "0" (zero). If the user chooses the option to enter the data required in a less structured way enter "1". All available details on the address of the recipient beneficial owner's agent or intermediary should then be presented as one left-justified string of 149 bytes, blank or "/" (slash) used as a delimiter between parts of the address. PLEASE NOTE that the address country code is outside this area of choosable format. 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.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreeForm7</td>
<td><em>(address free format)</em></td>
<td>1324</td>
<td>149</td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

If the user has entered "1" in field #50, he has opted to enter the data about the address of the recipient beneficial owner's agent or intermediary in one long left-justified field of 149 bytes. Enter the street and number, the city, the country sub-entity - if applicable (state, province or other political subunit) - and the postal (zip) code. Use blank as a delimiter between parts of the address. You may also use "/" (slash) as a delimiter to designate a new line in the printed form of the address. Do not enter the address country code in this field. It is strongly recommended that the address information be presented in the way the state where the recipient beneficial owner's agent or intermediary resides or is located usually
structures the address. Refer to the structure information to which the link "ad-strct" points - depending on implementation. Leave blank if not applicable or no data are available.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>FixForm7</td>
<td><em>(address fixed format)</em></td>
<td></td>
<td></td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

The details of the subdivision in fields are given below.

If the user has entered "0" (zero) in field #50, he has opted to enter the data about the address of the recipient beneficial owner's agent or intermediary in four separate fields (#51 through #54). Leave blank if not applicable or no data are available.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F051</td>
<td>Street</td>
<td>1324</td>
<td>70</td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #50, he has opted to enter the data about the address of the recipient beneficial owner's agent or intermediary in four separate fields (#51 through #54). In field #51, enter the complete name of the street and the building number where the recipient beneficial owner's agent or intermediary resides or is situated. It is strongly recommended that the address information be presented in the way the state where the recipient beneficial owner's agent or intermediary resides or is located usually structures the address. Refer to the structure information to which the link "ad-strct" points - depending on implementation.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F052</td>
<td>City</td>
<td>1394</td>
<td>35</td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #50, he has opted to enter the data about the address of the recipient beneficial owner's agent or intermediary in four separate fields (#51 through #54). In field #52, enter the name of the city or municipality where the recipient beneficial owner's agent or intermediary resides or is situated.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F053</td>
<td>CountrySubentity</td>
<td>1429</td>
<td>35</td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #50, he has opted to enter the data about the address of the recipient beneficial owner's agent or intermediary in four separate fields (#51 through #54). In field #53, enter the name of the country sub-entity if applicable in which the city where the recipient beneficial owner's agent or intermediary resides or is situated. Sub-entities may include states, regions, laender, provinces, or other political subunits. Leave blank if not applicable or no data are available.
If the user has entered "0" (zero) in field #50, he has opted to enter the data about the address of the recipient beneficial owner's agent or intermediary in four separate fields (#51 through #54). In field #54, enter the postal (zip) code of the address of the recipient beneficial owner's agent or intermediary. It is strongly recommended that the address information be presented in the way the state where the recipient beneficial owner's agent or intermediary resides or is located usually structures the address. Refer to the structure information to which the link "ad-strct" points - depending on implementation. Leave blank if not applicable or no data are available.

<table>
<thead>
<tr>
<th>Field/Field Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F055</td>
<td><strong>CountryCode</strong></td>
<td>1473</td>
<td>2</td>
<td>a</td>
<td>ISO 3166 ; blank if unknown</td>
</tr>
</tbody>
</table>

Enter the ISO 3166 two-byte alpha country code of the address entered in bytes 1324 through 1472. This field is not included in the alternatively fixed or free area in order to provide a unique location for this information.

<table>
<thead>
<tr>
<th>Field/Field Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>APR</td>
<td>***ActualPayer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This is the source of the payment that is described in this record, as determined by or documented to the sending country.

<table>
<thead>
<tr>
<th>Field/Field Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F056</td>
<td><strong>CountryCodeForTIN1</strong></td>
<td>1475</td>
<td>2</td>
<td>a</td>
<td>ISO 3166 ; blank if unknown</td>
</tr>
</tbody>
</table>

If a TIN is entered in field #57, enter the ISO 3166 two-byte alpha country code of the state that issued the TIN. The country code of the sending country is the most likely entry here, although any country code may be indicated.

<table>
<thead>
<tr>
<th>Field/Field Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F057</td>
<td><strong>TIN1</strong></td>
<td>1477</td>
<td>20</td>
<td>an</td>
<td>Actual Payer Tax Identification Number or Blank if data not available</td>
</tr>
</tbody>
</table>

If applicable, enter the TIN that the country identified in country code field #56 has issued to the actual payer.

<table>
<thead>
<tr>
<th>Field/Field Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F058</td>
<td><strong>CountryCodeForTIN2</strong></td>
<td>1497</td>
<td>2</td>
<td>a</td>
<td>ISO 3166 ; blank if unknown</td>
</tr>
</tbody>
</table>
If a TIN is entered in field #59, enter the ISO 3166 two-byte alpha country code of the state that issued the TIN.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F059</td>
<td><strong>TIN2</strong></td>
<td>1499</td>
<td>20</td>
<td>an</td>
<td>Actual Payer Tax Identification Number or Blank if data not available</td>
</tr>
</tbody>
</table>

If applicable, enter the TIN that the country identified in country code field #58 has issued to the actual payer. There is no rule whatsoever, if TINs assigned by two countries to the same recipient agent or intermediary are known, which TIN to enter as TIN1 and which as TIN2. However, the two numbers should not be identical (i.e., do not enter the same number both in fields #57 and #59).

Enter the appropriate type of the payer by choosing out of the codelist #1.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F060</td>
<td><strong>OECDPayerCode</strong></td>
<td>1519</td>
<td>2</td>
<td>an</td>
<td>See Codelist Table 1</td>
</tr>
</tbody>
</table>

This is the documented name of the actual payer.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>N_APR</td>
<td><strong>(name)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For fields #62 through #65, the user has the option to enter the data about the name of the actual payer either as one long left-justified field of 210 bytes in positions 1522 through 1731, or to spread the data over 4 fields. If the user chooses the option to enter the data required in fields #62 through #65 in one field, enter "1". This will be the common case with legal entities. All available name details should then be entered as one left-justified string of 210 bytes, blanks serving as delimiter between name parts. If the user chooses the option to enter the data required in fields #62 through #65 in separate fields, enter "0" (zero). The free format should be used if there is any doubt about the name details (which is keyname, which is first name etc.) as the receiving country may be in a better position to make a correct judgment. Note that "keyname" is intended to be a culturally neutral notion for what otherwise might be called "surname", "last name", "family name" etc.
If the user has entered "1" in field #61 he has opted to enter the data about the name of the actual payer in one long left-justified field of 210 bytes. For individuals enter the family name, the first name, middle name or initials (if available), the title (if available, e.g. Mr., Mrs.), and/or the suffix (if available, e.g., Esquire, Senior). For legal entities (corporations, partnerships, etc.), enter the registered or documented business name. Use blank as a delimiter between parts of the name.

If the user has entered "0" (zero) in field #61, he has opted to enter the data about the name of the actual payer in four separate fields. In fields #62 to #65, enter for individuals the family name (#62), the first name, middle name and/or initials (if available) (#63), the title (if available, e.g. Mr., Mrs.) (#64) and/or the suffix (if available, e.g. Esquire, Senior) (#65). Note that "keyname" is intended to be a culturally neutral notion for what otherwise might be called "surname", "last name", "family name" etc. For legal entities (corporations, partnerships, etc.) if you do not use free form as recommended, enter the complete business name in field #62.

If the user has entered "0" (zero) in field #61, he has opted to enter the data about the name of the actual payer in four separate fields. In field #62 enter the family name for individuals or the business name for legal entities (if you do not use free form as recommended for legal entities).

If the user has entered "0" (zero) in field #61, he has opted to enter the data about the name of the actual payer in four separate fields. In field #63 enter the first name, the middle name and/or the initials. Leave blank in the case of a legal entity.

Title includes: Mr, Mrs, Doctor, Sir, Professor etc.
If the user has entered "0" (zero) in field #61, he has opted to enter the data about the name of the actual payer in four separate fields. In field #64, enter the title. Titles may include Mr; Mrs; Doctor, Sir, Professor, etc. Abbreviations may be used as customary. Leave blank if no data are available or in the case of a legal entity.

<table>
<thead>
<tr>
<th>Field/Field Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F065</td>
<td>Suffix</td>
<td>1697</td>
<td>35</td>
<td>an</td>
<td>Suffix includes: Esquire, Senior, Junior etc.</td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #61, he has opted to enter the data about the name of the actual payer in four separate fields. In field #65, enter the suffix. Suffixes may include Esquire, Senior, Junior, etc. Abbreviations may be used as customary. Leave blank if no data are available or in the case of a legal entity.

<table>
<thead>
<tr>
<th>Field/Field Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>A_APR</td>
<td><strong>(Address)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In this group of fields enter the address information of the actual payer.

<table>
<thead>
<tr>
<th>Field/Field Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F066</td>
<td>*AddrFormatType</td>
<td>1732</td>
<td>1</td>
<td>n</td>
<td>0 fixed; 1 free; If free, all available address details will be given as a left justified string in bytes 1733 through 1881</td>
</tr>
</tbody>
</table>

For fields #67 through #70, the user has the option to enter the data about the address of the actual payer either as one long field or to spread the data over four fields. If the user chooses the option to enter the data required in fields #67 through #70 in separate fields, enter "0" (zero). If the user chooses the option to enter the data required in a less structured way enter "1". All available details on the address of the actual payer should then be presented as one left-justified string of 149 bytes, blank or "/" (slash) used as a delimiter between parts of the address. PLEASE NOTE that the address country code is outside this area of choosable format. 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 adress.

<table>
<thead>
<tr>
<th>Field/Field Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreeForm9</td>
<td>*(address free format)</td>
<td>1733</td>
<td>149</td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

If the user has entered "1" in field #66, he has opted to enter the data about the address of the actual payer in one long left-justified field of 149 bytes. Enter the street and number, the city, the country sub-entity - if applicable (state, province or other political subunit) - and the postal (zip)
The details of the subdivision in fields are given below.

If the user has entered "0" (zero) in field #66, he has opted to enter the data about the address of the actual payer in four separate fields (#67 through #70). In field #67, enter the complete name of the street and the building number where the actual payer resides or is situated.

If the user has entered "0" (zero) in field #66, he has opted to enter the data about the address of the actual payer in four separate fields (#67 through #70). In field #68, enter the name of the city or municipality where the actual payer resides or is situated.

If the user has entered "0" (zero) in field #66, he has opted to enter the data about the address of the actual payer in four separate fields (#67 through #70). In field #69, enter the name of the country sub-entity if applicable in which the city where the actual payer resides or is situated. Sub-entities may include states, regions, laender, provinces, or other political subunits. Leave blank if no data are available.

If the user has entered "0" (zero) in field #66, he has opted to enter the data about the address of the actual payer in four separate fields (#67 through #70). In field #70, enter the postal (zip) code of the address of the actual payer. Leave blank if no data are available.

If the user has entered "0" (zero) in field #66, he has opted to enter the data about the address of the actual payer in four separate fields (#67 through #70). In field #67, enter the complete name of the street and the building number where the actual payer resides or is situated.
<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>_<em><strong><strong><strong><strong><strong><strong><strong>Field/Group_Name</strong></strong></strong></strong></strong></strong></strong></em></th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F071</td>
<td><strong>CountryCode</strong></td>
<td>1882</td>
<td>2</td>
<td>a</td>
<td>ISO 3166 ; blank if unknown</td>
</tr>
</tbody>
</table>

Enter the ISO 3166 two-byte alpha country code of the address entered in bytes 1733 through 1881. This field is not included in the alternatively fixed or free area in order to provide a unique location for this information.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>_<em><strong><strong><strong><strong><strong><strong><strong>Field/Group_Name</strong></strong></strong></strong></strong></strong></strong></em></th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAI</td>
<td>***Payer Agent Or Intermediary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This is a party through which the actual payer has effected the payment. If no such party is identified, the entire field group should be left blank. This even applies to those fields that would have to be numeric in the "existence" case.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>_<em><strong><strong><strong><strong><strong><strong><strong>Field/Group_Name</strong></strong></strong></strong></strong></strong></strong></em></th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F072</td>
<td>**CountryCodeForTIN1</td>
<td>1884</td>
<td>2</td>
<td>a</td>
<td>ISO 3166 ; blank if unknown</td>
</tr>
</tbody>
</table>

If a TIN is entered in field #73, enter the ISO 3166 two-byte alpha country code of the state that issued the TIN. Note that the agent or intermediary may reside or do business anywhere, therefore this can be any country.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>_<em><strong><strong><strong><strong><strong><strong><strong>Field/Group_Name</strong></strong></strong></strong></strong></strong></strong></em></th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F073</td>
<td>**TIN1</td>
<td>1886</td>
<td>20</td>
<td>an</td>
<td>If applicable enter a TIN that the country with country code field#72 has issued for the agent or intermediary.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>_<em><strong><strong><strong><strong><strong><strong><strong>Field/Group_Name</strong></strong></strong></strong></strong></strong></strong></em></th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F074</td>
<td>**CountryCodeForTIN2</td>
<td>1906</td>
<td>2</td>
<td>a</td>
<td>ISO 3166 ; blank if unknown</td>
</tr>
</tbody>
</table>

If a TIN is entered in field #75, enter the ISO 3166 two-byte alpha country code of the state that issued the TIN. Note that the agent or intermediary may reside or do business anywhere, therefore this can be any country.
If applicable, enter the TIN that the country identified in country code field #74 has issued to the actual payer's agent or intermediary. There is no rule whatsoever, if TINs assigned by two countries to the same recipient agent or intermediary are known, which TIN to enter as TIN1 and which as TIN2. However, the two numbers should not be identical (i.e., do not enter the same number both in fields #73 and #75).

This is the documented name of the actual payer's agent or intermediary. In fields #76 through #80, the user should enter data about the name of the actual payer's agent or intermediary if applicable. Refer to those fields for more information.

For fields #77 to #80, the user has the option to enter the data about the name of the actual payer's agent or intermediary either as one long left-justified field of 210 bytes in positions 1929 through 2138, or to spread the data over 4 fields. If the user chooses the option to enter the data required in fields #77 to #80 in one field, enter "1". This will be the common case with legal entities. All available name details should then be entered as one left-justified string of 210 bytes, blanks serving as delimiter between name parts. If the user chooses the option to enter the data required in fields #77 to #80 in separate fields, enter "0" (zero). The free format should be used if there is any doubt about the name details (which is keyname, which is first name etc.) as the receiving country may be in a better position to make a correct judgment. Note that "keyname" is intended to be a culturally neutral notion for what otherwise might be called "surname", "last name", "family name" etc. To allow for the whole "payer's agent or intermediary" group of fields to be left blank if no such information is known, this field is defined alphanumeric in contrast to the "name" group of fields and a blank is a valid entry in the case mentioned.

If the user has entered "1" in field #76 he has opted to enter the data about the name of the actual payer's agent or intermediary in one long left-justified field of 210 bytes. For individuals enter the family name, the first name, middle name or initials (if available), the title (if available,
Mr., Mrs., and/or the suffix (if available, e.g., Esquire, Senior). For legal entities (corporations, partnerships, etc.), enter the registered or documented business name. Use blank as a delimiter between parts of the name. Leave blank if no data are available.

The details of the subdivision in fields are given below.

Field/Field Group | Field/Group Name | Start | Length | Data Type | Remarks
--- | --- | --- | --- | --- | ---
FixForm10 | *(name fixed format)* | | | | The details of the subdivision in fields are given below.

If the user has entered "0" (zero) in field #76, he has opted to enter the data about the name of the actual payer's agent or intermediary in four separate fields. In fields #77 to #80, enter for individuals the family name (#77), the first name, middle name and/or initials (if available) (#78), the title (if available, e.g. Mr., Mrs.) (#79) and/or the suffix (if available, e.g. Esquire, Senior) (#80). Note that "keyname" is intended to be a culturally neutral notion for what otherwise might be called "surname", "last name", "family name" etc. For legal entities (corporations, partnerships, etc.) if you do not use free form as recommended, enter the complete business name in field #77.

Field/Field Group | Field/Group Name | Start | Length | Data Type | Remarks
--- | --- | --- | --- | --- | ---
F077 | Keyname | 1929 | 70 | an | Keyname is the family name for individuals and the business name for legal entities.

If the user has entered "0" (zero) in field #76, he has opted to enter the data about the name of the actual payer's agent or intermediary in four separate fields. In field #77 enter the family name for individuals or the business name for legal entities (if you do not use free form as recommended for legal entities).

Field/Field Group | Field/Group Name | Start | Length | Data Type | Remarks
--- | --- | --- | --- | --- | ---
F078 | OtherNames | 1999 | 70 | an | Other name includes first name, middle name and/or initials.

If the user has entered "0" (zero) in field #76, he has opted to enter the data about the name of the actual payer's agent or intermediary in four separate fields. In field #78, enter the first name, the middle name and/or the initials. Leave blank if no data are available or in the case of a legal entity.

Field/Field Group | Field/Group Name | Start | Length | Data Type | Remarks
--- | --- | --- | --- | --- | ---
F079 | Title | 2069 | 35 | an | Title includes: Mr, Mrs, Doctor, Sir, Professor etc.

If the user has entered "0" (zero) in field #76, he has opted to enter the data about the name of the actual payer's agent or intermediary in four separate fields. In field #79, enter the title. Titles may include Mr; Mrs; Doctor, Sir, Professor, etc. Abbreviations may be used as customary. Leave blank if no data are available or in the case of a legal entity.
If the user has entered "0" (zero) in field #76, he has opted to enter the data about the name of the actual payer's agent or intermediary in four separate fields. In field #80, enter the suffix. Suffixes may include Esquire, Senior, Junior, etc. Abbreviations may be used as customary. Leave blank if no data are available or in the case of a legal entity.

In this group of fields, enter address information about the actual payer's agent or intermediary.

For fields #82 through #85, the user has the option to enter the data about the address of the actual payer's agent or intermediary either as one long field or to spread the data over four fields. If the user chooses the option to enter the data required in fields #82 through #85 in separate fields, enter "0" (zero). If the user chooses the option to enter the data required in a less structured way enter "1". All available details on the address of the actual payer's agent or intermediary should then be presented as one left-justified string of 149 bytes, blank or "/" (slash) used as a delimiter between parts of the address. PLEASE NOTE that the address country code is outside this area of choosable format. 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.

If the user has entered "1" in field #81, he has opted to enter the data about the address of the actual payer's agent or intermediary in one long left-justified field of 149 bytes. Enter the street and number, the city, the country sub-entity - if applicable (state, province or other political subunit) - and the postal (zip) code. Use blank as a delimiter between parts of the address. You may also use "/" (slash) as a delimiter to designate a new line in the printed form of the address. Do not enter the address country code in this field. It is strongly recommended that the address information be
presented in the way the state where the actual payer's agent or intermediary resides or is located usually structures the address. Refer to the structure information to which the link "ad-strct" points - depending on implementation. Leave blank if not applicable or no data are available.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>FixForm11</td>
<td><em>(address fixed format)</em></td>
<td></td>
<td></td>
<td></td>
<td>The details of the subdivision in fields are given below.</td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #81, he has opted to enter the data about the address of the actual payer's agent or intermediary in four separate fields (#82 through #85). Leave blank if not applicable or no data are available.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F082</td>
<td>Street</td>
<td>2140</td>
<td>70</td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #81, he has opted to enter the data about the address of the actual payer's agent or intermediary in four separate fields (#82 through #85). In field #82, enter the complete name of the street and the building number where the actual payer's agent or intermediary resides or is situated. It is strongly recommended that the address information be presented in the way the state where the actual payer's agent or intermediary resides or is located usually structures the address. Refer to the structure information to which the link "ad-strct" points - depending on implementation. Leave blank if not applicable or no data are available.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F083</td>
<td>City</td>
<td>2210</td>
<td>35</td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #81, he has opted to enter the data about the address of the actual payer's agent or intermediary in four separate fields (#82 through #85). In field #83, enter the name of the city or municipality where the actual payer's agent or intermediary resides or is situated. Leave blank if not applicable or no data are available.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F084</td>
<td>CountrySubentity</td>
<td>2245</td>
<td>35</td>
<td>an</td>
<td></td>
</tr>
</tbody>
</table>

If the user has entered "0" (zero) in field #81, he has opted to enter the data about the address of the actual payer's agent or intermediary in four separate fields (#82 through #85). In field #84, enter the name of the country sub-entity if applicable in which the city where the actual payer's agent or intermediary resides or is situated. Sub-entities may include states, regions, laender, provinces, or other political subunits. Leave blank if not applicable or no data are available.
If the user has entered "0" (zero) in field #81, he has opted to enter the data about the address of the actual payer's agent or intermediary in four separate fields (#82 through #85). In field #85, enter the postal (zip) code of the address of the actual payer's agent or intermediary. It is strongly recommended that the address information be presented in the way the state where the actual payer's agent or intermediary resides or is located usually structures the address. Refer to the structure information to which the link "ad-strct" points - depending on implementation. Leave blank if not applicable or no data are available.

Enter the ISO 3166 two-byte alpha country code of the address entered in bytes 2140 through 2288. This field is not included in the alternatively fixed or free area in order to provide a unique location for this information.

This is the end of the tax year in the source country. The date to be entered should refer to the tax year in which the income payment was effected. The reason for the inclusion of this field into the record is mainly that in some countries the tax year does not coincide with the calendar year. The data must be provided left-justified with four digits for the year and optionally two digits for the month and another two digits for the day of the month, refer to ISO 8601.

This group of fields contains the data about the payment that is the subject of this record. 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) As a rule the following equations should hold: field#94 (NIP) = field#92 (GIP) - field#96 (TWH); field#96 (TWH) = field#92 (GIP) * field#97 (TR). 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 country specific part of this manual, the technical notes accompanying the file and/or in field#104, depending on the scope of the exception. It is assumed that field#99 (tax refund) is not bound to the other amounts by an equation of the above kind.
### **Field/Field Group**

<table>
<thead>
<tr>
<th>Field/Field Group</th>
<th><strong>Field/Group_Name</strong></th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>F088</strong></td>
<td><strong>DateOfPayment</strong></td>
<td>2299</td>
<td>8</td>
<td>an</td>
<td>ISO 8601: CCYYMMDD or CCYYMM or CCYY or Blank if unknown</td>
</tr>
<tr>
<td><strong>OECDPaymentType</strong></td>
<td></td>
<td>2307</td>
<td>4</td>
<td>an</td>
<td>See Codelist Table 2</td>
</tr>
<tr>
<td><strong>SendingCountryPaymentType</strong></td>
<td></td>
<td>2311</td>
<td>4</td>
<td>an</td>
<td></td>
</tr>
<tr>
<td><strong>GrossIncomePaid</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Fields #91 and #92 contain data about the Gross Income Paid.</td>
</tr>
</tbody>
</table>

Enter the actual date the income was paid. Note the remark on "payment" made in the introduction to this field group (identifier PD). If necessary specific information concerning the date of payment can be included in the country specific part of this manual, the technical notes accompanying the file and/or in field #104. Enter the data left-justified with four digits for the year and optionally two digits for the month and another two digits for the day of the month, refer to ISO 8601. Leave blank if no data are available.

Codelist 2 contains the codes for the various types of income that may be paid to a recipient beneficial owner. Only these codes should be used, as they correspond to the articles of the OECD Model Double Taxation Convention and must be chosen in accordance with the definition given in those articles. Enter the code corresponding to the type of income represented by the amount in field #92. Note that code #15, corresponding to "dependent personal services" has been extended to allow a better description of this type of income. Additional information necessary should be entered in the filler fields at the end of the record (preferably field #103). At present, this standard does not impose rules on how to do this, but leaves this implementation detail for agreement between the countries involved. If a sending country feels that no code listed in codelist 2 corresponds to the type of income being reported, use code #21 (other income) and provide an explanation in field #104.

The OECD standard provides common definitions for most income types, thus, deviation from use of the OECD codes given in codelist 2 is discouraged. However, if a sending country is unable to use the OECD codes to describe the income type, it may use its own categories. In this case, the sending country should identify country-specific payment types in this field. The definition of such country-specific payment types must be conveyed to the receiving country by adequate means, preferably in the country-specific part of this manual. Field #90 can also be used in addition to an entry in field #89 in order to make a better distinction. In this case, too, separate information as to the meaning of the codes used is necessary.
Field/Field_Group | Field/Group_Name | Start | Length | Data_Type | Remarks
--- | --- | --- | --- | --- | ---
F091 | *CurrencyCode | 2315 | 3 | a | ISO 4217-3 Alpha

Enter the three-byte ISO code for the currency of the gross income. This should be the currency in which the payment has actually been effected. Leave blank to indicate that you do not provide a meaningful amount in field #92 (this allows you to enter a meaningless 0 - zero - and not violate the numeric field type). Never leave blank if there is an amount in field #92, as the information is meaningless without this code.

Field/Field_Group | Field/Group_Name | Start | Length | Data_Type | Remarks
--- | --- | --- | --- | --- | ---
F092 | *Amount-MajorUnitOfCurrency | 2318 | 18 | n | Major units of currency only

This entry must be strictly numerical. Examples: $4000 would be formally wrong as the currency has to be identified by the ISO code in field #91. 4000.71 would be formally wrong as it contains a decimal point and minor currency units. 798,000 would be formally wrong as no editing characters are allowed in a strictly numerical field. 40050 for an amount of 400 dollars and 50 cents would be semantically wrong as it would have to be interpreted as 40050 dollars according to the definition of this standard. The decision to round or truncate to a major currency unit is left to the discretion of the sending country, information about the procedure followed should be included in the country specific part of this manual.

Field/Field_Group | Field/Group_Name | Start | Length | Data_Type | Remarks
--- | --- | --- | --- | --- | ---
F093 | *CurrencyCode | 2336 | 3 | a | ISO 4217-3 Alpha

Enter the three-byte ISO code for the currency of the net income paid. This should be the currency in which the payment has actually been effected. Leave blank to indicate that you do not provide a meaningful amount in field #96 (this allows you to enter a meaningless 0 - zero - and not violate the numeric field type). Never leave blank if there is an amount in field #96, as the information is meaningless without this code.

Field/Field_Group | Field/Group_Name | Start | Length | Data_Type | Remarks
--- | --- | --- | --- | --- | ---
F094 | *Amount-MajorUnitOfCurrency | 2339 | 18 | n | Major units of currency only

This entry must be strictly numerical. Examples: $4000 would be formally wrong as the currency has to be identified by the ISO code in field #91. 4000.71 would be formally wrong as it contains a decimal point and minor currency units. 798,000 would be formally wrong as no editing characters are allowed in a strictly numerical field. 40050 for an amount of 400 dollars and 50 cents would be semantically wrong as it would have
to be interpreted as 40050 dollars according to the definition of this standard. The decision to round or truncate to a major currency unit is left to
the discretion of the sending country, information about the procedure followed should be included in the country specific part of this manual.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>TWH</td>
<td>**TaxWithheld</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fields #95 and #96 contain data about the tax withheld.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F095</td>
<td>*CurrencyCode</td>
<td>2357</td>
<td>3</td>
<td>a</td>
<td>ISO 4217-3 Alpha</td>
</tr>
</tbody>
</table>

Enter the three-byte ISO code for the currency of the tax withheld. This should be the currency in which the payment has actually been effected. Leave blank to indicate that you do not provide a meaningful amount in field #96 (this allows you to enter a meaningless 0 - zero - and not violate the numeric field type). Never leave blank if there is an amount in field #96, as the information is meaningless without this code.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F096</td>
<td>*Amount-MajorUnitOfCurrency</td>
<td>2360</td>
<td>18</td>
<td>n</td>
<td>Major units of currency only</td>
</tr>
</tbody>
</table>

This entry must be strictly numerical. Examples: $4000 would be formally wrong as the currency has to be identified by the ISO code in field #91. 4000.71 would be formally wrong as it contains a decimal point and minor currency units. 798,000 would be formally wrong as no editing characters are allowed in a strictly numerical field. 40050 for an amount of 400 dollars and 50 cents would be semantically wrong as it would have to be interpreted as 40050 dollars according to the definition of this standard. The decision to round or truncate to a major currency unit is left to the discretion of the sending country, information about the procedure followed should be included in the country specific part of this manual.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F097</td>
<td>**TaxRate</td>
<td>2378</td>
<td>4</td>
<td>n</td>
<td>or blank</td>
</tr>
</tbody>
</table>

Enter the withholding tax rate actually applied to the payment. If used, this must be strictly numerical. The digits entered are always interpreted as hundredth of percents. E.g., 5 percent will be given as 0500, 1550 is a tax rate of 15.5 percent. Blank is a legal entry (and the only legal non-numeric entry) denoting the fact that no tax rate is given.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRF</td>
<td>**TaxRefund</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Enter tax refunded, if any, after the amount entered in field #96 was withheld. If more than one refund is effected with respect to the income of this record enter the sum of these refunds.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F098</td>
<td>*CurrencyCode</td>
<td>2382</td>
<td>3</td>
<td>a</td>
<td>ISO 4217-3 Alpha</td>
</tr>
</tbody>
</table>

Enter the three-byte ISO code for the currency of the tax refunded. This should be the currency in which the payment has actually been effected. Leave blank to indicate that you do not provide a meaningful amount in field #99 (this allows you to enter a meaningless 0 - zero - and not violate the numeric field type). Never leave blank if there is an amount in field #99, as the information is meaningless without this code.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F099</td>
<td>*Amount-MajorUnitOfCurrency</td>
<td>2385</td>
<td>18</td>
<td>n</td>
<td>Major units of currency only</td>
</tr>
</tbody>
</table>

This entry must be strictly numerical. Examples: $4000 would be formally wrong as the currency has to be identified by the ISO code in field #91. 4000.71 would be formally wrong as it contains a decimal point and minor currency units. 798,000 would be formally wrong as no editing characters are allowed in a strictly numerical field. 40050 for an amount of 400 dollars and 50 cents would be semantically wrong as it would have to be interpreted as 40050 dollars according to the definition of this standard. The decision to round or truncate to a major currency unit is left to the discretion of the sending country, information about the procedure followed should be included in the country specific part of this manual.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F100</td>
<td>*Date Of Refund Or First Refund</td>
<td>2403</td>
<td>8</td>
<td>an</td>
<td>ISO 8601: CCYMMDD or CCYYMM or CCYY or Blank if unknown</td>
</tr>
</tbody>
</table>

Enter the actual date the refund was made. If the amount entered in field #99 reflects more than one refund, enter only the date of the first refund. Enter the data left-justified with four digits for the year and optionally two digits for month and another two digits for the day of the month. Refer to ISO 8601. Leave blank if no data are available.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F101</td>
<td>***Sender´sReference</td>
<td>2411</td>
<td>70</td>
<td>an</td>
<td>Sender´s number for any queries relating to the particular record</td>
</tr>
</tbody>
</table>

Enter in this field a unique reference "number" (not necessarily strictly numeric) identifying each record transmitted. Even in the case of a repetition (see field #1) this reference must be distinct from the one of the record repeated. All communication about the record should refer to the content of this field. It is best to make this reference "all time unique" for the sending country and not to rely on additional information (tax year,
date of transmission, etc.) not included in the reference to identify the record. It is also advisable to build the reference in a way that links it to the internal records of the sending country.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F102</td>
<td>***CorrectionReference</td>
<td>2481</td>
<td>70</td>
<td>an</td>
<td>Relates to the Sender’s number if this record is a correction or repetition record (see field 1). Otherwise blank.</td>
</tr>
</tbody>
</table>

Be sure to include this reference in the case of correction or repetition of a record sent before, as otherwise correct handling of the record will not be guaranteed or may even be impossible.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F103</td>
<td>***FillerGeneralUse</td>
<td>2551</td>
<td>105</td>
<td>an</td>
<td>Blank unless used. Details to be provided in technical notes accompanying the transmittal.</td>
</tr>
</tbody>
</table>

As a rule this filler field is reserved for changes decided upon by an OECD official procedure. You may also use this field in a way you describe in the technical notes you give to the receiving country at the time of the transmittal, but you should do so only if field #104 does not suffice. The use of field #103 is recommended in connection with the extended OECD payment type 15b, though there is no firm rule how to do this and a bilateral agreement or documentation in technical notes is therefore necessary.

<table>
<thead>
<tr>
<th>Field/Field_Group</th>
<th>Field/Group_Name</th>
<th>Start</th>
<th>Length</th>
<th>Data_Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>F104</td>
<td>***Filler Specific Arrangements</td>
<td>2656</td>
<td>105</td>
<td>an</td>
<td>Blank unless used. Details agreed between sending country and recipient country and specified in technical notes accompanying the transmittal.</td>
</tr>
</tbody>
</table>

The use of this field is completely at the discretion of the sending country as long as it is made clear to the receiving country what information is reflected.
Codelist1: Type of recipient and type of payer

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Individual</td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>Corporation</td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>Partnership</td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>Business organisation other than corporation or partnership</td>
<td></td>
</tr>
<tr>
<td>05</td>
<td>Government or international organisation</td>
<td></td>
</tr>
<tr>
<td>06</td>
<td>Other (specify in filler)</td>
<td></td>
</tr>
<tr>
<td>07</td>
<td>Unknown</td>
<td>Whenever the type of a party is unknown, code 07 has to be specified. Do not leave the field blank.</td>
</tr>
</tbody>
</table>

Codelist2: Type of income code corresponding to the numbering of the Articles of the OECD Model Convention of Income and Capital

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>06</td>
<td>Income from immovable property</td>
<td></td>
</tr>
<tr>
<td>07</td>
<td>Business profits</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Dividends</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Interest</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Royalties</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Capital Gains</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Income from independent personal services</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Income from dependent personal services</td>
<td></td>
</tr>
<tr>
<td>15a</td>
<td>Gross amount (including fringe benefits)</td>
<td></td>
</tr>
<tr>
<td>15b</td>
<td>Money amount only but additional information on fringe benefits to follow in fillers according to</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>bilateral agreements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15c</td>
<td>Money amount only</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Directors' fees</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Income derived from activities of an artist or sportsman</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Pensions</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Income from government services and public pensions</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Payments to students for education and training</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Other income</td>
<td></td>
</tr>
</tbody>
</table>
OECD STANDARDISED TRANSMISSION 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 (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.

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)

---

4 Refer to Council Recommendations C(92)50/FINAL (23 July 1992) and C(97)30/FINAL (10 July 1997).
and is based on extensible markup language (XML), 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)

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.

---

6 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**

<table>
<thead>
<tr>
<th></th>
<th>On-request and spontaneous exchange</th>
<th>Automatic exchange</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Physical exchange</strong></td>
<td>Email attachment</td>
<td>Magnetic tape, diskette, CD Rom (or DVD) transmitted via normal mail, commercial courier, diplomatic bag etc.</td>
</tr>
<tr>
<td><strong>Electronic exchange</strong></td>
<td>Email attachment</td>
<td>Magnetic tape, diskette, CD Rom (or DVD) transmitted via normal mail, commercial courier, diplomatic bag etc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Level 1</strong></th>
<th><strong>Transmission media</strong></th>
<th><strong>Level 2</strong></th>
<th><strong>Encryption</strong></th>
<th><strong>Level 3</strong></th>
<th><strong>Content Format</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Paper documents transmitted via normal mail, commercial courier, diplomatic bag etc.</strong></td>
<td><strong>No encryption of paper documents.</strong></td>
<td><strong>Standard Transmission Encryption</strong> [see DAFFE/CFA/ WP8/TIES(2003)5/CONF for encryption and key management procedures]</td>
<td><strong>N/A for paper documents.</strong></td>
<td><strong>PDF, JPG or TIFF files for scanned documents. RTF for electronically stored documents.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Email attachment</strong></td>
<td><strong>Standard Transmission Encryption</strong> (see footnote 13)</td>
<td><strong>Standard Transmission Encryption</strong> (see footnote 13)</td>
<td><strong>PDF, JPG or TIFF files for scanned documents. RTF for electronically stored documents.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Magnetic tape, diskette, CD Rom (or DVD) transmitted via normal mail, commercial courier, diplomatic bag etc.</strong></td>
<td><strong>Standard Transmission Format (SMF); or Standard Transmission Format (STF)</strong></td>
<td><strong>Standard Transmission Format (SMF); or Standard Transmission Format (STF)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

7 DVDs are not generally used by competent authorities but can hold much more data than CD-ROMs.
8 Automatic exchange via email is possible (subject to file size) but is generally not considered necessary.
9 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).
10 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.
11 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 version="1.0" encoding="UTF-8"?>
  <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:sequence>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
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).

Collection of general STF datatypes: Schema stftypes

<?xml version="1.0" encoding="UTF-8"?>
  <!-- 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>

<html>
<header>
<title>Collection of general STF datatypes: Schema stftypes</title>
</header>
<body>
<pre>
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).

Collection of general STF datatypes: Schema stftypes

<?xml version="1.0" encoding="UTF-8"?>
  <!-- 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>
</pre>
</body>
</html>
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

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

This is a datatype for an attribute to an address. It serves to indicate the legal character of that address (residential, business etc.)

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 Pediatrics, Inc. 'SMFAliasOrOther' should be chosen if the document is generated from a legacy SMF record, where no further distinction is possible.
Kind of the Identifier that is provided for a party

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.

Kind of the Qualifier that describes a Payment Type

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.

Qualifier for a Payment: Gross or Net Income, Tax Withheld or Refunded

Qualifier for a Payment: Gross or Net Income, Tax Withheld or Refunded
<xsd:complexType 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>

<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>

<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>

<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>
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.

```xml
<xsd:complexType>
  <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 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'.

```xml
<xsd:complexType>
  <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:choice>
  </xsd:sequence>
</xsd:complexType>
```
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.

Sender's unique identifier of this document

Reference id of the message of the document referred to if this is repetition or correction

Reference id of the document referred to if this is repetition or correction

Data (other than Name and Address) to describe and identify an Individual

Data (other than Name and Address) to describe and identify a Legal Entity

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.

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: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"/>
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.

<xs:annotation>
  <xs:documentation xml:lang="en">His Excellency, Estate of the Late ...</xs:documentation>
</xs:annotation>
<xs:element name="PrecedingTitle" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>His Excellency, Estate of the Late ...</xs:documentation>
  </xs:annotation>
  <xs:complexType mixed="true"/>
</xs:element>
<xs:element name="Title" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>Greeting title. Example: Mr, Dr, Ms, Herr, etc. Can have multiple titles.</xs:documentation>
  </xs:annotation>
  <xs:complexType mixed="true"/>
</xs:element>
<xs:element name="FirstName" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
    <xs: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.</xs:documentation>
  </xs:annotation>
  <xs:complexType mixed="true">
    <xs:attribute name="xnlNameType">
      <xs:annotation>
        <xs: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.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
</xs:element>
<xs:element name="MiddleName" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
    <xs: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.</xs:documentation>
  </xs:annotation>
  <xs:complexType mixed="true">
    <xs:attribute name="xnlNameType">
      <xs:annotation>
        <xs:documentation>Defines the name type of Middle Name. Example: First name, middle name, maiden name, father's name, given name, etc.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
</xs:element>
<xs:element name="NamePrefix" minOccurs="0">
  <xs:annotation>
    <xs:documentation>de, van, van de, von, etc. Example: Derick de Clarke</xs:documentation>
  </xs:annotation>
  <xs:complexType mixed="true">
    <xs:attribute name="xnlNameType">
      <xs:annotation>
        <xs: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.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
</xs: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, Surname, 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">
            ...
        </xsd:documentation>
    </xsd:annotation>
</xsd:complexType>
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.

Other Security Identification Number: A Security Identification Number other than the standard ISIN, the attribute to indicate the kind of such number.

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.

Other Info 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.

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: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
    </xsd:documentation>
  </xsd:annotation>
</xsd:complexType>
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.

```xml
taxRefundTypeType
    <xsd:complexType name="taxRefundTypeType">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">To indicate the tax refund, the date of the last day of the relevant tax year is given.</xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="taxRefundDate" type="xsd:date" /
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

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

```xml
isotypes_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="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
- ...<br/>
- ZM -- ZAMBIA
- ZW -- ZIMBABWE
</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>
</xsd:schema>
```
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:restriction base="xsd:string">
  <xsd:enumeration value="AED"/>
  <xsd:enumeration value="AFA"/>
  ...
  <xsd:enumeration value="ZMK"/>
  <xsd:enumeration value="ZWD"/>
</xsd:restriction>`

`<xsd:simpleType>`

### `oecdtypes_v1.xsd`

`<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
    </xsd:documentation>
  </xsd:annotation>
</xsd:schema>`
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>
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 exchange.

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.
In XML schema terms this is expressed as

```xml
<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.

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 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 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:oecd:ties:stf:v1);

This results in the structure shown in figure 3.
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.

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 \[ \square \] 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:

```
  DocSpec
    DocTypeIndic
    DocRefId
    CorMessageRefId
    CorDocRefId
```

*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:
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:

```xml
<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="SpecificPaymentType_Type">
            <xsd:attribute name="specificPaymentTypeQlf" type="xsd:string" fixed="Dummy"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

Figure 6: Data about the Income
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 specificities_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 specificities_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 born 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>

The payment qualifier (paymentQlf) 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:choice>
  </xsd:sequence>
</xsd:complexType>
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 ’acctNoQlf’ and ‘secNoQlf’ 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:

![Diagram of Party_Type]

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:

```xml
<xsd:enumeration value="SMFAliasOrOther"/>
<xsd:enumeration value="indiv"/>
```
and for addresses:

<xs:enumeration value="residentialOrBusiness"/>
<xs:enumeration value="residential"/>
<xs:enumeration value="business"/>
<xs: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:

<xs:complexType name="PartyId_Type">
  <xs:simpleContent>
    <xs:extension base="xsd:string">
      <xs:attribute name="partyIdType" type="partyIdType_Type" use="required"/>
      <xs:attribute name="issuedBy" type="xsd:string" use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs: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:
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](image)

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 breakdown 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:
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:
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-discriminable 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:

![Structured Address in STF](image)

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
<?xml-stylesheet type="text/xsl" href="pppp/smff2stf01.2.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:

```xml
<!-- ********************************************************************* -->
<!-- *************   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

```xml
<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>
```

```xml
<MessageSpec>
  <SendingCountry>GB</SendingCountry>
  <ReceivingCountry>US</ReceivingCountry>
  <Warning>Please do not use this</Warning>
</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
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 )

8.5 Examples of elements of Party type

A beneficial owner Party element representing an individual person

A payer Party element (address example adapted from an OASIS CIQ standard example)
8.6 Examples of PaymentData elements

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

```xml
<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.

```xml
<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:

```xml
...<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) </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
        <xsd:extension base="SpecificPaymentType_Type">
            <xsd:attribute name="specificPaymentTypeQlf" type="xsd:string" fixed="UK2001"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

- S20 interest from extremely large deposits
- ...
- P47a fees for directors of chains of nightclubs
- ...
8.7 A complete message containing two documents for different tax years, one a correction to another document assumed to be sent before

```xml
<STF_OECD xmlns="urn:oecd: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">
      <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 xmlnsNameType="Given Name">Mary</FirstName>
          <MiddleName xmlnsNameType="Middle Initial">R</MiddleName>
          <NamePrefix xmlnsNameType="Prefix">de</NamePrefix>
          <LastName xmlnsNameType="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>
      <CountryCode>DE</CountryCode>
      <AddressFix>
        <Street>Friedhofstrasse 1</Street>
        <PostCode>53225</PostCode>
        <City>Bonn</City>
      </AddressFix>
      <Address/>
      <PersData>
        <IndivPersData>
          <Gender>F</Gender>
          <Nationality>FR</Nationality>
          <BirthDate>1937-08-13</BirthDate>
          <BirthCity>Paris</BirthCity>
          <BirthCitySubentity>Montmartre</BirthCitySubentity>
        </IndivPersData>
      </PersData>
    </RecipientBeneficialOwner>
    <RecipientAgentOrIntermediary oecdLegalType="01">
    </RecipientAgentOrIntermediary>
  </STF_DIRECT>
</STF_OECD>
```
<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">
  <PartyId partyIdType="TIN" issuedBy="US">999999999</PartyId>
  <Name nameType="legal">
    <NameFree>Grey Dancers Great Performances</NameFree>
  </Name>
  <Address legalAddressType="business">
    <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>
</STF_DIRECT>
<RecipientBeneficialOwner oecdLegalType="03">
  <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">
  <PartyId partyIdType="TIN" issuedBy="US">124534</PartyId>
  <Name nameType="legal">
    <NameFree>First Banking for Nothing</NameFree>
  </Name>
  <Address legalAddressType="unspecified">
    <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>
  </Payment>
</PaymentData>
</STF_DIRECT>
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)
BRIDGING PROGRAMME FROM SMF TO STF

<?xml version="1.0" encoding="UTF-8"?>
<!-- DISCLAIMER:
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. -->

<xsl:stylesheet version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:output method="xml"/>
 <!--************************************************************************ -->
 <!--************************************************************************ -->
 <!-- **********              Main Template ********** -->
 <!-- ********** -->
 <!--************************************************************************ -->
 <!--************************************************************************ -->
 <xsl:template match="/">
 <STF_OECD xmlns="urn:oecd:ties:stf:v1"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 </xsl:template>
</xsl:stylesheet>
<xsl:for-each select="SMFFile/Record">
  <xsl:call-template name="record"/>
</xsl:for-each>
</STF_OECD>
<xsl:template name="record">
  <STF_DIRECT xmlns="urn:oecd:ties:stf:v1">
    <DocSpec>
      <DocTypeIndic>
        <xsl:value-of select="substring(.,1,1)"/>
      </DocTypeIndic>
      <DocRefId>
        <xsl:value-of select="normalize-space(substring(.,2411, 70))"/>
      </DocRefId>
      <CorrDocRefId>
        <xsl:value-of select="normalize-space(substring(.,2481, 70))"/>
      </CorrDocRefId>
    </DocSpec>
    <RecipientBeneficialOwner oecdLegalType="{substring(.,46,2)}">
      <ResCountryCode>
        <xsl:value-of select="substring(.,2,2)"/>
      </ResCountryCode>
      <PartId partyIdType="TIN" issuedBy="{substring(.,2,2)}">
        <xsl:value-of select="normalize-space(substring(.,4, 20))"/>
      </PartId>
    </RecipientBeneficialOwner>
  </STF_DIRECT>
</xsl:template>
<PartyId partyIdType="TIN" issuedBy="{substring(.,24,2)}">
  <xlsl:value-of select="normalize-space(substring(.,26,20))"/>
</PartyId>
</xsl:if>
</Name>
<xsl:variable name="RecType">
  <xsl:value-of select="substring(.,46,2)"/>
</xsl:variable>
<xsl:if test="$RecType=01">
  <xsl:attribute name="nameType">indiv</xsl:attribute>
</xsl:if>
<xsl:if test="$RecType='02' or $RecType='03' or $RecType='04' or $RecType='05'">
  <xsl:attribute name="nameType">legal</xsl:attribute>
</xsl:if>
<xsl:call-template name="stfname">
  <xsl:with-param name="begin">56</xsl:with-param>
</xsl:call-template>
</Name>
<xsl:if test="substring(.,340,1)!=' '">
  <Name nameType="SMFAliasOrOther">
    <xsl:call-template name="stfname">
      <xsl:with-param name="begin">340</xsl:with-param>
    </xsl:call-template>
  </Name>
</xsl:if>
<Address>
  <xsl:attribute name="legalAddressType"><xsl:if test="substring(.,762,1)=0">residentialOrBusiness</xsl:if><xsl:if test="substring(.,762,1)=1">registeredOffice</xsl:if><xsl:if test="substring(.,762,1)=2">unspecified</xsl:if></xsl:attribute>
  <xsl:call-template name="stfaddress">
    <xsl:with-param name="begin">763</xsl:with-param>
  </xsl:call-template>
</Address>
<xsl:if test="substring(.,915,1)!=' '">
  <Address>
    <xsl:attribute name="legalAddressType"><xsl:if test="substring(.,915,1)=0">residentialOrBusiness</xsl:if><xsl:if test="substring(.,915,1)=1">registeredOffice</xsl:if><xsl:if test="substring(.,915,1)=2">unspecified</xsl:if></xsl:attribute>
    <xsl:call-template name="stfaddress">
      <xsl:with-param name="begin">916</xsl:with-param>
    </xsl:call-template>
  </Address>
</xsl:if>
<xsl:if test="substring(.,267,1)='F' or substring(.,267,1)='M'">
  <PersData>
    <IndivPersData>
      <Gender>
        <xsl:value-of select="substring(.,267,1)"/>
      </Gender>
      <xsl:if test="substring(.,48,8)!=' '">
        <BirthDate>
          <xsl:call-template name="stfdate">
          </xsl:call-template>
        </BirthDate>
      </xsl:if>
    </IndivPersData>
  </PersData>
</xsl:if>
<ActualPayer oecdLegalType="{substring(.,1519,2)}">
<xsl:if test="substring(.,1475,22)!='                      '">
  <PartyId partyIdType="TIN" issuedBy="{substring(.,1475,2)}">
    <xsl:value-of select="normalize-space(substring(.,1477,20))"/>
  </PartyId>
</xsl:if>
<xsl:if test="substring(.,1497,22)!='                      '">
  <PartyId partyIdType="TIN" issuedBy="{substring(.,1497,2)}">
    <xsl:value-of select="normalize-space(substring(.,1499,20))"/>
  </PartyId>
</xsl:if>
{Name>
<xsl:variable name="RecType">
  <xsl:value-of select="substring(.,1519,2)"/>
</xsl:variable>
<xsl:if test="$RecType=01">
  <xsl:attribute name="nameType">indiv</xsl:attribute>
</xsl:if>
<xsl:if test="$RecType='02' or $RecType='03' or $RecType='04' or $RecType='05'"> 
  <xsl:attribute name="nameType">legal</xsl:attribute>
</xsl:if>
<xsl:call-template name="stfname">
  <xsl:with-param name="begin">1521</xsl:with-param>
</xsl:call-template>
</Name>
<Address>
<xsl:call-template name="stfaddress">
  <xsl:with-param name="begin">1732</xsl:with-param>
</xsl:call-template>
</Address>
</ActualPayer>

<PayerAgentOrIntermediary oecdLegalType="07">
<xsl:if test="substring(.,1928,1)!=' '">
  <PartyId partyIdType="TIN" issuedBy="{substring(.,1884,2)}">
    <xsl:value-of select="normalize-space(substring(.,1886,20))"/>
  </PartyId>
</xsl:if>
</PayerAgentOrIntermediary>
<Payment paymentQlf="twh">
  <PaymentDate>
    <xsl:call-template name="stfdate">
      <xsl:with-param name="begin">2299</xsl:with-param>
    </xsl:call-template>
  </PaymentDate>
  <MonAmnt currCode="{substring(.,2357,3)}">
    <xsl:value-of select="substring(.,2360,18)"/>
  </MonAmnt>
  <TaxRate>
    <xsl:value-of select="substring(.,2378,2)"/>
    <xsl:value-of select="substring(.,2380,2)"/>
  </TaxRate>
</Payment>

<xsl:if test="substring(.,2382,3)!='   '"><<<br />
  <Payment paymentQlf="trf">
    <PaymentDate>
      <xsl:call-template name="stfdate">
        <xsl:with-param name="begin">2403</xsl:with-param>
      </xsl:call-template>
    </PaymentDate>
    <MonAmnt currCode="{substring(.,2382,3)}">
      <xsl:value-of select="substring(.,2385,18)"/>
    </MonAmnt>
    <xsl:if test="substring(.,2388,4)!='    '"><<<br />
      <TaxRate>
        <xsl:value-of select="substring(.,2389,2)"/>
        <xsl:value-of select="substring(.,2391,2)"/>
      </TaxRate>
    </xsl:if>
  </Payment>
</xsl:if>
</PaymentData>
<!-- ***************************    OtherInfo
************************************* -->
<OtherInfo>
  <xsl:if test="substring(.,2551,10)!='          '"><<<br />
    <ContentOfSMFFillerGeneralUse>
      <xsl:value-of select="substring(.,2551,105)"/>
    </ContentOfSMFFillerGeneralUse>
  </xsl:if>
  <xsl:if test="substring(.,2656,10)!='          '"><<<br />
    <ContentOfSMFFillerSpecificArrangements>
      <xsl:value-of select="substring(.,2656,105)"/>
    </ContentOfSMFFillerSpecificArrangements>
  </xsl:if>
</OtherInfo>
</xsl:template>
</xsl:template>
BRIDGING PROGRAMME FROM STF TO SMF

<?xml version="1.0" encoding="UTF-8"?>
<!--
This xslt stylesheet transforms an "SFT_OECD" document into a text file
conforming to SMF record layout.
The SFT_OECD layout here referred is as the XML Schema "stfdirect-0.3.4.xsd"
with types schema files pasted in (standard XSLT cannot process included
files).

Author: Sergio Baldelli - Ministry of Economy and Finance - Italy -
sergio.baldelli@finanze.it
--> AL 1104: supported version of stfdirect is stfdirect-1.0.xsd now. -->
<!-- Parts that are surrounded by comments marked "AL 1104" have been
changed, removed or added by Arndt Liesen in November 2004 -->
<!-- AL 1104 DISCLAIMER:
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. -->
<!-- ATTENTION: changes must be done in order to match the latest version of
STF:
- OtherInfo
- RBO TINs
- ... -->
<!-- AL 1104: The above has been taken care of as far as it seemed necessary
--><xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmns:stf="urn:oecd:ties:stf:v1">
  <xsl:output method="text" indent="no" omit-xml-declaration="yes"
encoding="UTF-8"/>
  <!-- ************* Global variable definition with SMF field length
  ******************** -->
  <!-- ************* OECD codes fields -->
  <!-- ************* Identification fields -->
  <!-- ************* Name fields -->
</xsl:stylesheet>
<xsl:variable name="otherNameLen" select="70"/>
<xsl:variable name="titleLen" select="35"/>
<xsl:variable name="suffixLen" select="35"/>
<xsl:variable name="nameFreeLen" select="210"/>

<-- Address fields
  --------------------------------------------------
  <xsl:variable name="addressTypeLen" select="1"/>
  <xsl:variable name="addressFormatTypeLen" select="1"/>
  <xsl:variable name="streetLen" select="70"/>
  <xsl:variable name="cityLen" select="35"/>
  <xsl:variable name="citySubEntityLen" select="35"/>
  <xsl:variable name="countrySubEntityLen" select="35"/>
  <xsl:variable name="postalCodeLen" select="9"/>
  <xsl:variable name="addressFreeLen" select="149"/>
  <xsl:variable name="countryCodeLen" select="2"/>

<-- ISO 3166 -->

<-- Record Information fields
  --------------------------------------------------
  <xsl:variable name="dataTypeIndicatorLen" select="1"/>

<-- SMF field #1 -->

<-- SMF field #101 & #102 -->

<-- SMF field #103 & #104 -->

<-- Tax fields
  --------------------------------------------------
  <xsl:variable name="taxYearEndLen" select="8"/>
  <xsl:variable name="paymentTypeLen" select="4"/>
  <xsl:variable name="currencyCodeLen" select="3"/>

<-- ISO 4217 -->

<-- payment and tax withheld or refund -->

<-- SMF record length
  --------------------------------------------------
  <xsl:variable name="recLen" select="2760"/>

<-- Whole party length excepted RBO -->

<-- Miscellaneous fields
  --------------------------------------------------
  <xsl:variable name="dateLen" select="8"/>
  <xsl:variable name="partyLen" select="407"/>

<-- ISO 8601 -->

<-- As above but for debugging -->

<-- Used to get white spaces to fill fields up to their length. NB The
star character (*) is necessary to avoid XSLT blank stripping (any other non
whitespace character would be OK). The star is skipped setting 2 the start
position of substring function. -->

<-- As above but for debugging -->

<xsl:variable name="zeros">000000000000000000</xsl:variable>

<!-- AL 1104 Begin -->
<xsl:template match="stf:STF_OECD">
  <xsl:for-each select="stf:STF_DIRECT">
    <!-- Processes STF equivalents of SMF records -->
    <xsl:value-of select="stf:DocSpec/stf:DocTypeIndic"/>
    <!-- SMF #1 -->
    <xsl:apply-templates select="stf:RecipientBeneficialOwner"/>
    <xsl:choose>
      <xsl:when test="stf:RecipientAgentOrIntermediary">
        <xsl:apply-templates select="stf:RecipientAgentOrIntermediary"/>
        <!-- SMF #41-55 -->
      </xsl:when>
      <xsl:otherwise>
        <!-- Fills with blanks -->
        <xsl:value-of select="substring($zeros, 2, $partyLen)"/>
        <!-- APr has OECD payer code -->
      </xsl:otherwise>
    </xsl:choose>
    <xsl:choose>
      <xsl:when test="stf:ActualPayer">
        <xsl:apply-templates select="stf:ActualPayer"/>
        <!-- SMF APr #56-71 -->
      </xsl:when>
      <xsl:otherwise>
        <!-- Fills with blanks -->
        <xsl:value-of select="substring($zeros, 2, $OECDrecipientPayerTypeLen + $partyLen)"/>
        <!-- APr has OECD payer code -->
      </xsl:otherwise>
    </xsl:choose>
    <xsl:choose>
      <xsl:when test="stf:PayerAgentOrIntermediary">
        <xsl:apply-templates select="stf:PayerAgentOrIntermediary"/>
        <!-- SMF APr #72-86 -->
      </xsl:when>
      <xsl:otherwise>
        <!-- Fills with blanks -->
        <xsl:value-of select="substring($zeros, 2, $partyLen)"/>
      </xsl:otherwise>
    </xsl:choose>
    <xsl:apply-templates select="stf:PaymentData"/>
    <!-- SMF #87-100 -->
    <!-- SMF #101 -->
  </xsl:for-each>
</xsl:template>

<!-- AL 1104 End -->
length(stf:DocSpec/stf:CorrDocRefId))}} />
  <!-- SMF #102 -->
  <xsl:value-of select="concat(stf:OtherInfo, substring($blanks, 2, $fillerLen - string-length(stf:OtherInfo)))" />
  <!-- SMF #103 -->
  <xsl:value-of select="substring($blanks, 2, $fillerLen)" />
  <!-- SMF #104 -->
</xsl:for-each>
</xsl:template>
<xsl:template match="stf:MessageSpec" />
  <!-- Avoid default rules -->
</xsl:template>

*************** RBO

*************** RBO

<xsl:template match="stf:RecipientBeneficialOwner" />
  <!-- AL 1104 begin -->
  <xsl:choose>
    <xsl:when test="stf:ResCountryCode">
    </xsl:when>
    <xsl:otherwise>
      <xsl:call-template name="TIN">
        <xsl:with-param name="partyIdNo">1</xsl:with-param>
        <xsl:with-param name="TINno">1</xsl:with-param>
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
  <!-- AL 1104 end -->
  <!-- SMF #6 -->
  <!-- STF date format: CCYY-MM-DD, SMF format CCYYMMDD: must strip dashes -->
  <xsl:value-of select="concat(@oecdLegalType, substring($blanks, 2, $OECDrecipientPayerTypeLen - string-length(@oecdLegalType)))" />
  <!-- AL 23.12.04 -->
<xsl:choose>
  <xsl:when test="stf:PersData/stf:IndivPersData/stf:Gender">
  </xsl:when>
  <xsl:otherwise>
    <xsl:text>U</xsl:text>
  </xsl:otherwise>
</xsl:choose>

<!-- SMF #13-->

<!-- SMF #14-->

<!-- SMF #15-->

<!-- SMF #16-->
<xsl:call-template name="RBOAliasOrOtherNames">
  <xsl:with-param name="nameNo" select="1"/>
</xsl:call-template>

<!-- 'In care of' name is not possible in STF: fills with blanks #22-26 -->
<xsl:value-of select="substring($blanks, 2, $nameFormatTypeLen + $nameFreeLen )"/>

<xsl:apply-templates select="stf:Address[1]"/>

<!-- SMF #27-33 -->
<xsl:choose>
  <xsl:when test="stf:Address[2]">
    <!-- RBO may have a second address. #34-40 -->
    <xsl:apply-templates select="stf:Address[2]"/>
  </xsl:when>
  <xsl:otherwise>
    <!-- Fills with blanks -->
    <xsl:value-of select="substring($blanks, 2, $addressTypeLen + $addressFreeLen + $addressFormatTypeLen + $countryCodeLen)"/>
  </xsl:otherwise>
</xsl:choose>

<!--[RAI, PAI, APR]-->

<xsl:template match="stf:RecipientAgentOrIntermediary | stf:ActualPayer | stf:PayerAgentOrIntermediary">
  <xsl:call-template name="TIN">
    <xsl:with-param name="partyIdNo">1</xsl:with-param>
    <xsl:with-param name="TINno">1</xsl:with-param>
  </xsl:call-template>
</xsl:template>
<xsl:if test="name() = 'ActualPayer'">
  <!-- APr has OECD payer code -->
  <xsl:value-of select="concat(@oecdLegalType, substring($blanks, 2, $OECDrecipientPayerTypeLen - string-length(@oecdLegalType)))"/>
</xsl:if>
<xsl:call-template name="NamesDefault">
  <xsl:with-param name="nameNo" select="1"/>
</xsl:call-template>
<xsl:apply-templates select="stf:Address[1]"/>
</xsl:template>
<!-- *************************** PD *********************************** -->
<xsl:template match="stf:PaymentData">
  <xsl:value-of select="concat(translate(stf:TaxYearEnd, '-', ''), substring($blanks, 2, $dateLen - string-length(translate(stf:TaxYearEnd, '-', ''))))"/>
</xsl:template>
<xsl:choose>
  <xsl:when test="stf:Payment[@paymentQlf = 'gip']/stf:PaymentDate">
    <xsl:value-of select="concat(translate(stf:Payment[@paymentQlf = 'gip']/stf:PaymentDate, '-', ''), substring($blanks, 2, $dateLen - string-length(translate(stf:Payment[@paymentQlf = 'gip']/stf:PaymentDate, '-', ''))))"/>
  </xsl:when>
  <xsl:otherwise>
    <xsl:value-of select="concat(translate(stf:Payment[@paymentQlf = 'nip']/stf:PaymentDate, '-', ''), substring($blanks, 2, $dateLen - string-length(translate(stf:Payment[@paymentQlf = 'nip']/stf:PaymentDate, '-', ''))))"/>
  </xsl:otherwise>
</xsl:choose>
<!-- AL23.12.04 changes for SMF#89 and SMF#90 due to different payment data structure -->
<xsl:value-of select="concat(substring(stf:OECDPaymentType,1,$paymentTypeLen), substring($blanks, 2, $paymentTypeLen - string-length(substring(stf:OECDPaymentType,1,$paymentTypeLen))))"/>
<xsl:choose>
  <xsl:when test="stf:SpecificPaymentType">
    <xsl:value-of select="concat(substring(stf:SpecificPaymentType,1,$paymentTypeLen), substring($blanks, 2, $paymentTypeLen - string-
    
116
length(substring(stf:SpecificPaymentType,1,$paymentTypeLen))))"/>
</xsl:when>
<xsl:otherwise>
<!-- No country specific payment type: fills with blank -->
<xsl:value-of select="substring($blanks, 2, $paymentTypeLen)"/>
</xsl:otherwise>
</xsl:choose>
<xsl:choose>
<!-- SMF #91-92 -->
<xsl:when test="stf:Payment[@paymentQlf = 'gip']">
<xsl:apply-templates select="stf:Payment[@paymentQlf = 'gip']"/>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="substring($blanks, 2, $currencyCodeLen +
$amountLen)"/>
</xsl:otherwise>
</xsl:choose>
</xsl:when>
<xsl:otherwise>
<!-- SMF #93-94 -->
<xsl:choose>
<xsl:when test="stf:Payment[@paymentQlf = 'nip']">
<xsl:apply-templates select="stf:Payment[@paymentQlf = 'nip']"/>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="substring($blanks, 2, $currencyCodeLen +
$amountLen)"/>
</xsl:otherwise>
</xsl:choose>
</xsl:when>
<xsl:otherwise>
<!-- SMF #95-96 -->
<xsl:choose>
<xsl:when test="stf:Payment[@paymentQlf = 'twh']">
<xsl:apply-templates select="stf:Payment[@paymentQlf = 'twh']"/>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="substring($blanks, 2, $currencyCodeLen +
$amountLen)"/>
</xsl:otherwise>
</xsl:choose>
</xsl:when>
<xsl:otherwise>
<!-- SMF #97 - Must manage format conversion from STF -->
<xsl:variable name="intDigit" select="concat(substring('0000 ', 1, 2 - string-length(substring-before(stf:Payment/stf:TaxRate, '.'))),
substring-before(stf:Payment/stf:TaxRate, '.'))"/>
<xsl:variable name="fracDigit" select="concat(substring-after(stf:Payment/stf:TaxRate, '.'),
substring('0000 ', 1, 2 - string-length(substring-after(stf:Payment/stf:TaxRate, '.'))))"/>
<xsl:value-of select="concat($intDigit, $fracDigit)"/>
</xsl:when>
<xsl:otherwise>
<!-- SMF #98-99 -->
<xsl:when test="stf:Payment[@paymentQlf = 'trf']">
<xsl:apply-templates select="stf:Payment[@paymentQlf = 'trf']"/>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="substring($blanks, 2, $currencyCodeLen +
$amountLen)"/>
</xsl:otherwise>
</xsl:when>
<xsl:otherwise>
<!-- SMF #100 -->
<xsl:value-of select="concat(translate(stf:Payment[@paymentQlf =
}
'trf']/stf:PaymentDate, '-' , '')', substring($blanks, 2, $dateLen - string-length(translate(stf:Payment[@paymentQlf = 'trf']/stf:PaymentDate, '-' , '''))))"/>

</xsl:template>

<!-- ************************************************************ -->
<xsl:template match="stf:Payment">
<xsl:value-of select="stf:MonAmnt/@currCode"/>
<!-- Must manage possible amount with trailing decimal -->
<xsl:choose>
  <xsl:when test="contains(stf:MonAmnt, '.')">
    <!-- AL 1104 <xsl:value-of select="concat(substring-before(stf:MonAmnt, '.'), substring($blanks, 2, $amountLen - string-length(substring-before(stf:MonAmnt, '.'))))"/> -->
    <!-- AL 1104 Begin -->
    <xsl:value-of select="concat( substring($zeros, 2, $amountLen - string-length(substring-before(stf:MonAmnt, '.'))),substring-before(stf:MonAmnt, '.')")"/>
    <!-- AL 1104 End -->
    </xsl:when>
  <xsl:otherwise>
    <!-- AL 1104 <xsl:value-of select="concat(stf:MonAmnt, substring($blanks, 2, $amountLen - string-length(stf:MonAmnt)))"/> -->
    <!-- AL 1104 Begin -->
    <xsl:value-of select="concat( substring($zeros, 2, $amountLen - string-length(stf:MonAmnt)),stf:MonAmnt)"/>
    <!-- AL 1104 End -->
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<!-- ************************************************************ -->
<!-- In SMF only 2 TINs are possible while in STF more than 2 PartyId may occur.
Thus for each SMF TIN field it is necessary to scan all the STF PartyId (starting
from the related SMF field position) in order to find PartyId of TIN type. If no such
field is filled with blanks
-->
<!-- AL 1104 Begin -->
<xsl:template name="TINrbo1">
<xsl:param name="partyIdNo"/>
</xsl:template>

<!-- ************************************************************ -->
<xsl:template match="stf:Payment">
<xsl:value-of select="stf:MonAmnt/@currCode"/>
<!-- Must manage possible amount with trailing decimal -->
<xsl:choose>
  <xsl:when test="contains(stf:MonAmnt, '.')">
    <!-- AL 1104 <xsl:value-of select="concat(substring-before(stf:MonAmnt, '.'), substring($blanks, 2, $amountLen - string-length(substring-before(stf:MonAmnt, '.'))))"/> -->
    <!-- AL 1104 Begin -->
    <xsl:value-of select="concat( substring($zeros, 2, $amountLen - string-length(substring-before(stf:MonAmnt, '.'))),substring-before(stf:MonAmnt, '.')")"/>
    <!-- AL 1104 End -->
    </xsl:when>
  <xsl:otherwise>
    <!-- AL 1104 <xsl:value-of select="concat(stf:MonAmnt, substring($blanks, 2, $amountLen - string-length(stf:MonAmnt)))"/> -->
    <!-- AL 1104 Begin -->
    <xsl:value-of select="concat( substring($zeros, 2, $amountLen - string-length(stf:MonAmnt)),stf:MonAmnt)"/>
    <!-- AL 1104 End -->
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<!-- ************************************************************ -->
<!-- In RBO TINs must be managed in a special way. While in the other parties the two first
PartyIds of TIN type are mapped respectively into the 1st and 2nd TIN field, in RBO the
physical order is not relevant. The PartyId (of type TIN) of which
the issuedBy attribute matches the residence country code must mapped in the first SMF TIN
field.
Refer to TIN template for all common features.
-->
<!-- AL 1104 Begin -->
<xsl:template name="TINrbo1">
<xsl:param name="partyIdNo"/>
</xsl:template>

118
<xsl:template name="NamesDefault">
  <xsl:param name="nameNo"/>
  <xsl:choose>
    <xsl:when test="stf:Name[number($nameNo)]">
      <!-- Still names to scan -->
      <xsl:choose>
        <xsl:when test="stf:Name[number($nameNo)][not(@nameType)]">
          <xsl:apply-templates select="stf:Name[number($nameNo)]"/>
        </xsl:when>
        <xsl:otherwise>
          <!-- Call itself on next element -->
          <xsl:call-template name="NamesDefault">
            <xsl:with-param name="nameNo" select="$nameNo + 1"/>
          </xsl:call-template>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <xsl:otherwise>
      <xsl:call-template name="NamesLegalOrIndiv">
        <xsl:with-param name="nameNo" select="1"/>
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<!-- ************************** NamesLegalOrIndiv
************************** -->
<!-- Tests all the Name elements and maps the 1st one with attribute = "legal" or "indiv".
Otherwise fills with blanks. -->
<xsl:template name="NamesLegalOrIndiv">
  <xsl:param name="nameNo"/>
  <xsl:choose>
    <xsl:when test="stf:Name[number($nameNo)]">
      <!-- Still names to scan -->
      <xsl:choose>
        <xsl:when test="stf:Name[number($nameNo)]/@nameType = 'legal'">
          <xsl:apply-templates select="stf:Name[number($nameNo)]"/>
        </xsl:when>
        <xsl:otherwise>
          <!-- Call itself on next element -->
          <xsl:call-template name="NamesLegalOrIndiv">
            <xsl:with-param name="nameNo" select="$nameNo + 1"/>
          </xsl:call-template>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <xsl:otherwise>
      <!-- Fills with blanks -->
      <xsl:apply-templates select="stf:Name[number($nameNo)]"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
<xsl:template name="RBONames">
  <xsl:param name="nameNo"/>
  <xsl:choose>
    <xsl:when test="stf:Name[number($nameNo)]">
      <xsl:choose>
        <xsl:when test="not(stf:Name[number($nameNo)]/@nameType) or (contains($mainName, stf:Name[number($nameNo)]/@nameType) and stf:Name[number($nameNo)]/@nameType != '')">
          <!-- contain() returns true with void string; in case of empty attribute -->
          <xsl:apply-templates select="stf:Name[number($nameNo)]"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:call-template name="RBONames">
            <!-- Call itself on next element -->
            <xsl:with-param name="nameNo" select="$nameNo + 1"/>
          </xsl:call-template>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <xsl:otherwise>
      <!-- no one Name element (if any) satisfies the test: fills with blanks -->
      <xsl:value-of select="substring($blanks, 2, $nameFormatTypeLen + $nameFreeLen )"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<!-- *********************** RBOAliasOrOther Names **************************** -->
<!-- Tests all the Name elements and maps the 1st one for which holds:  
there is a nameType attribute AND its value is not a void string AND 
it is one of the following values: "aka", "dba", "nick", "alias", 
"SMFAliasOrOther". Otherwise fills with blanks. -->
<xsl:template name="RBOAliasOrOtherNames">
  <xsl:param name="nameNo"/>
  <xsl:choose>
    <xsl:when test="stf:Name[number($nameNo)]">
      <xsl:choose>
        <xsl:when test="stf:Name[number($nameNo)]/@nameType and stf:Name[number($nameNo)]/@nameType != '' and contains($alias, stf:Name[number($nameNo)]/@nameType)"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="substring($blanks, 2, $nameFormatTypeLen + $nameFreeLen )"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
<!-- Call itself on next element -->
<xsl:call-template name="RBOAliasOrOtherNames">
  <xsl:with-param name="nameNo" select="$nameNo + 1"/>
</xsl:call-template>
</xsl:otherwise>
</xsl:choose>

<xsl:when>
  <xsl:otherwise>
    <!-- no one Name element (if any) satisfies the test: fills with blanks -->
  </xsl:otherwise>
</xsl:choose>

<xsl:otherwise>
  <!-- no one Name element (if any) satisfies the test: fills with blanks -->
  <xsl:value-of select="substring($blanks, 2, $nameFormatTypeLen + $nameFreeLen )"/>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

<!-- **************************** Name ********************************** -->
<xsl:template match="stf:Name">
  <xsl:choose>
    <xsl:when test="stf:NameFree">
      <xsl:text>1</xsl:text>
      <!-- SMF Name format type flag -->
      <xsl:choose>
        <!-- At birth name must not be appended to Alias name -->
        <xsl:when test="contains($alias, @nameType)">
          <xsl:value-of select="concat(./stf:NameFree,
            substring($blanks, 2, $nameFreeLen - string-length(./stf:NameFree)))"/>
        </xsl:when>
        <xsl:otherwise>
          <!-- Not an alias name: must manage at birth -->
          <xsl:call-template name="atbirthFix"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <xsl:otherwise>
      <!-- Not a free format name -->
      <xsl:call-template name="namefix"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<!-- ************************* Fix format name ************************* -->
<xsl:template name="namefix">
  <xsl:text>0</xsl:text>
  <!-- SMF Name format type flag -->
  <xsl:choose>
    <!-- At birth name must not be appended to Alias name -->
    <xsl:when test="contains($alias, @nameType)">
      <xsl:value-of select="concat(../stf:LastName ,
        substring($blanks, 2, $keyNameLen - string-length(../stf:LastName)))"/>
    </xsl:when>
    <xsl:otherwise>
      <!-- Not an alias name: must manage at birth -->
      <xsl:call-template name="atbirthFix"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<!-- Appends STF FirstName and MiddleName into SMF Other Name field -->
<xsl:variable name="otherNameBuff" select="normalize-space(concat(.//stf:FirstName, ' ', .//stf:MiddleName))"/>
<xsl:value-of select="concat($otherNameBuff, substring($blanks, 2, $otherNameLen - string-length($otherNameBuff)))"/>
<!-- Appends into SMF Suffix field STF GenerationIdentifier, Suffix and GeneralSuffix -->
<xsl:value-of select="concat(.//stf:Title, substring($blanks,2, $titleLen - string-length(.//stf:Title)))"/>
<xsl:variable name="suffixBuff" select="normalize-space(concat(.//stf:GenerationIdentifier, ' ', .//stf:Suffix, ' ', .//stf:GeneralSuffix))"/>
<xsl:value-of select="concat($suffixBuff, substring($blanks,2, $suffixLen - string-length($suffixBuff)))"/>
</xsl:template>

<!-- ************************ "at birth" (free form) ************************** -->
<xsl:template name="atbirthFree">
<xsl:choose>
<xsl:when test="following-sibling::stf:Name[@nameType='atbirth']/stf:NameFree">
<xsl:variable name="buff" select="normalize-space(concat(./stf:NameFree, ' at birth: ', following-sibling::stf:Name[@nameType='atbirth']/stf:NameFree))"/>
<xsl:value-of select="concat($buff, substring($blanks, 2, $nameFreeLen - string-length($buff)))"/>
</xsl:when>
<xsl:when test="preceding-sibling::stf:Name[@nameType='atbirth']/stf:NameFree">
<xsl:variable name="buff" select="normalize-space(concat(./stf:NameFree, ' at birth: ', preceding-sibling::stf:Name[@nameType='atbirth']/stf:NameFree))"/>
<xsl:value-of select="concat($buff, substring($blanks, 2, $nameFreeLen - string-length($buff)))"/>
</xsl:when>
<xsl:when test="following-sibling::stf:Name[@nameType='atbirth']/stf:NameFix">
<xsl:variable name="buff" select="normalize-space(concat(./stf:NameFree, ' at birth: ', following-sibling::stf:Name[@nameType='atbirth']/stf:NameFix/stf:LastName))"/>
<xsl:value-of select="concat($buff, substring($blanks, 2, $nameFreeLen - string-length($buff)))"/>
</xsl:when>
<xsl:when test="preceding-sibling::stf:Name[@nameType='atbirth']/stf:NameFix">
<xsl:variable name="buff" select="normalize-space(concat(./stf:NameFree, ' at birth: ', preceding-sibling::stf:Name[@nameType='atbirth']/stf:NameFix/stf:LastName))"/>
<xsl:value-of select="concat($buff, substring($blanks, 2, $nameFreeLen - string-length($buff)))"/>
</xsl:when>
<xsl:otherwise>
<!-- No sibling name with at birth attribute: copies the bare name -->
<xsl:value-of select="concat(./stf:NameFree, substring($blanks, 2, $nameFreeLen - string-length(./stf:NameFree)))"/>
</xsl:otherwise>
</xsl:choose>
(fix form)
<xsl:template match="stf:Address">
    <xsl:if test="parent::stf:RecipientBeneficialOwner">
        <!-- Only RBO has "Address type" field -->
        <xsl:choose>
            <xsl:when test="contains('stf:residentialOrBusiness', @legalAddressType)">
                <!-- AL 23.12.04 -->
                <xsl:text>0</xsl:text>
            </xsl:when>
            <xsl:when test="contains('stf:registeredOffice', @legalAddressType)">
                <!-- AL 23.12.04 -->
                <xsl:text>1</xsl:text>
            </xsl:when>
            <xsl:otherwise>
                <xsl:text>2</xsl:text>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:if>
    <xsl:choose>
        <xsl:when test="*[2] = stf:AddressFree">
            <!-- Second child = AddressFree ? -->
            <xsl:text>1</xsl:text>
            <xsl:value-of select="concat(./stf:AddressFree, substring($blanks, 2, $addressFreeLen - string-length(./stf:AddressFree)))"/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:text>0</xsl:text>
            <xsl:apply-templates select="stf:AddressFix"/>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>

<!-- ************************** Fix format address **************************** -->

<xsl:template match="stf:AddressFix">
    <!-- Appends into SMF Street field STF elements: street, building, suite and floor -->
    <xsl:variable name="streetBuff" select="normalize-space(concat(.//stf:Street, ' ', .//stf:BuildingIdentifier, ' ', .//stf:SuiteIdentifier, ' ', .//stf:FloorIdentifier))"/>
    <xsl:value-of select="concat($streetBuff, substring($blanks, 2, $streetLen - string-length($streetBuff)))"/>
    <xsl:value-of select="concat(stf:City, substring($blanks, 2, $cityLen - string-length(stf:City)))"/>
    <xsl:value-of select="concat(stf:CountrySubentity, substring($blanks, 2, $countrySubEntityLen - string-length(stf:CountrySubentity)))"/>
</xsl:template>