eCDF

Plateforme électronique de Collecte des Données Financières

DOCUMENTATION OF THE XML FILE eCDF VERSION V2.0

CENTRE DES TECHNOLOGIES DE L'INFORMATION DE L'ÉTAT

(24/11/2022)

Table of contents

1. Purpose of this document

eCDF can accept transfers of financial data in an XML file with a predefined structure. These XML files must be generated from the eCDF interface of an accounting tool, ensuring that financial data does not have to be entered twice and that the files are generated in the correct format required by eCDF.

After authorization by the CTIE, the developer gets access to an area dedicated to interface developers that contains all the information needed to develop this interface (XML documentation, XML schema, test feature ...) as well as an interface identifier that must be added to all the files generated by the interface to be developed.

The developer can himself do structure tests on his generated files, and when he thinks that the developed interface has been validated by a significant number of positive tests on generated XML files for different types of declarations, the developer may make a request to the eCDF team of the CTIE in order to officially validate and activate the interface for data transfers.

Following legal or technical changes, the specification of the files transferred into eCDF may evolve. Such an evolution of the specification results in a new eCDF-XML file version. In this case, the interface developer has to make evolutions to his interface in order to be compliant to the new version and he has to accomplish a new interface validation procedure with the CTIE. After a transition phase that cannot exceed three months and during which certain old forms can still be transferred, the old XML version will be deactivated and files created by XML interfaces which have not yet gotten the validation for the new version, will not be accepted by eCDF anymore.

If in this documentation the terms XML 1.1 and XML 2.0 are used, this does not refer to a version of the official XML specification, but to the version of the XML file format for eCDF.

2. Evolutions in the eCDF file version 2.0

As from 4.1.2021, the filing of annual accounts for financial years starting in 2020 or later must be done using as a new element the "mapping table", which explains how the values from the chart of accounts (PCN) are reported on the (abridged) balance sheet and the (abridged) profit and loss account.

In order to be able to make such filings using a "mapping table", new XML elements have become necessary and the XML 1.1 file has evolved into a new version 2.0, usable from 4.1.2021, which mainly brings the following changes, applicable only to annual accounts filings for financial years beginning in 2020 or later:

- new «MappingTable» element in the annual accounts filing;
- new «DeclarationGroup» element allowing to group together the elements of an annual accounts filing.

Version 2.0 of the XML file for eCDF still allows the transfer of

- annual accounts for the financial years 2011-2019, without using the «DeclarationGroup» XML element and without using a mapping table.
- all VAT declarations available in eCDF.

2.1. Addition of the mapping table in the annual accounts deposit

As of 4.1.2021, the mapping table is a new additional element in the annual accounts deposits for financial years starting in 2020 or later. The mapping table explains how the field values from the standard chart of accounts (PCN) are reported into the (abridged) balance sheet and the (abridges) profit and loss accounts.

The “Commission des Normes Comptables (CNC)” has provided a standard mapping table and for deposits done according to this standard mapping table, it’s sufficient to include in the XML file a « MappingTable »

XML element with the indication that the (abridged) balance sheet and the (abridges) profit and loss accounts have been calculated according to this standard mapping table.

This standard transition table is available on eCDF in the "Forms / Mapping table" menu. This page also offers a downloadable XML version of the standard mapping table.

Declarers having more advanced needs may use a custom mapping table in which they define their own calculations for the (abridged) balance sheet and the (abridges) profit and loss accounts. Of course, all these elements must constitute a coherent whole and the resulting (abridged) balance sheet and the (abridges) profit and loss accounts must be valid according to the rules specific to their own forms.

If a custom mapping table is used, the XML file must specify in detail the calculations that are different from the standard mapping table calculations. Calculations that are unchanged from the standard mapping table must not be listed and repeated in the custom mapping table.

2.2. Grouping of annual accounts elements in a « DeclarationGroup »

The new element « MappingTable » establishes a logical link between the chart of accounts, the (abridged) balance sheet and the (abridged) profit and loss account and these four elements must be consistent. In an XML 2.0 file, this logical link is expressed by grouping the elements of an annual accounts filing into a new XML element « DeclarationGroup ». This « DeclarationGroup » element must be used in annual accounts filings for financial years beginning in 2020 or later, consisting of more than one element, in order to group these elements.

2.2.1. Use case with « DeclarationGroup »

As from 4.1.2021, for the filing of annual accounts for financial years starting in 2020 or later, consists in most cases of the following four elements, which must be consistent:

- the “Standard chart of account”
- the “Balance sheet” or “Abridged Balance Sheet”
- the “Profit and loss account” or “Abridged profit and loss account”
- the “Mapping table”

For these filings, the four elements («Declaration»; «MappingTable») must be in the same XML file and be grouped by a new XML element «DeclarationGroup» to make it clear that they belong together and are consistent.

2.2.2. Use case without « DeclarationGroup »

As from 4.1.2021, for the filing of annual accounts for financial years starting in 2020 or later, there are a limited number of companies that are in an exceptional situation and are authorised to file in eCDF only their standard chart of accounts (PCN). This authorisation must be indicated in the PCN by the corresponding check box "Non-automated generation of the balance sheet and the profit and loss account" at the end of the form.

In this case, since the standardised chart of accounts (PCN) is the only element for these annual accounts to be filed through the XML file, it must not be placed in a « DeclarationGroup » element.

It is recalled that the new mode of operation using a mapping table and the elements grouped by the new XML element « DeclarationGroup » only applies to annual accounts filings for financial years starting in 2020 or later, if they consist of more than one element.

Annual accounts for financial years starting before 2020 must also be transferred through an XML file in the new 2.0 format, but continue to be transferred as individual declarations, without grouping by the «DeclarationGroup» element and without using a mapping table.

VAT returns are always considered as separate declarations and can never be located in a «DeclarationGroup» element, regardless of the year.

2.3. Annual Accounts 2020, model="1" and model="2"

Between 1.1.2020 and 31.12.2020, annual accounts declarations for financial years starting in 2020 may be filed in an XML 1.1 file as individual declarations. These declarations must contain the model="1".

```
<Declaration type="CA_BILAN" model="1" language="FR">
```

As of 04.01.2021, the annual accounts declarations for financial years starting in 2020 must be filed in an XML 2.0 file, using the mapping table, all declarations containing the model="2" and the whole grouped by the element « DeclarationGroup ».

```
<Declaration type="CA_BILAN" model="2" language="FR">
```

Example: filing of annual accounts for a shortened financial year from 1.1.2020 to 31.09.2020.

If the filing is made in 2020, the transfer must be made in an XML file 1.1, in which the three elements PCN, Balance Sheet and Profit and Loss Account are to be inserted as individual declarations (model="1").

If the filing is made later than 4.1.2021, the transfer must be made in an XML 2.0 file, in which a «DeclarationGroup» element groups the three declarations PCN (with additional checkboxes available in model="2"), (abridged) Balance Sheet and (abridged) Profit and Loss Account (model="2") together with the mapping table.

The eCDF website also contains a "news" entry with a document about the changes in the forms for 2020, which also explains the difference between model="1" and model="2".

Since the new concepts, «MappingTable» and «DeclarationGroup» legally required as of 2021 for annual accounts filings for financial years starting in 2020 or later, do not exist in XML 1.1 format, this XML 1.1 format can no longer be used from 2021 onwards, to transfer 2020 annual accounts. The transfer of other types of declarations that existed in XML 1.1, including VAT 2020 declarations, will remain available until 31.03.2021, when XML 1.1 will be definitively discontinued.

As of 4.1.2021, the standard XML file format for eCDF will be 2.0.

It should be noted that in XML 2.0, the annual accounts declarations for 2021 and subsequent years always start again with model="1". The model number of a declaration is only incremented if there is a substantial change to this form.

2.4. Schematic comparison of XML versions 1.1 and 2.0 with validity dates

This chapter contains a schematic comparison of XML versions 1.1 and 2.0 at different moments in time.

In the following illustration, declarer AAAAAA represents the standard case whereas declarer BBBBBBB represents the case of a declarer having received the exceptional authorization to file only the PCN.

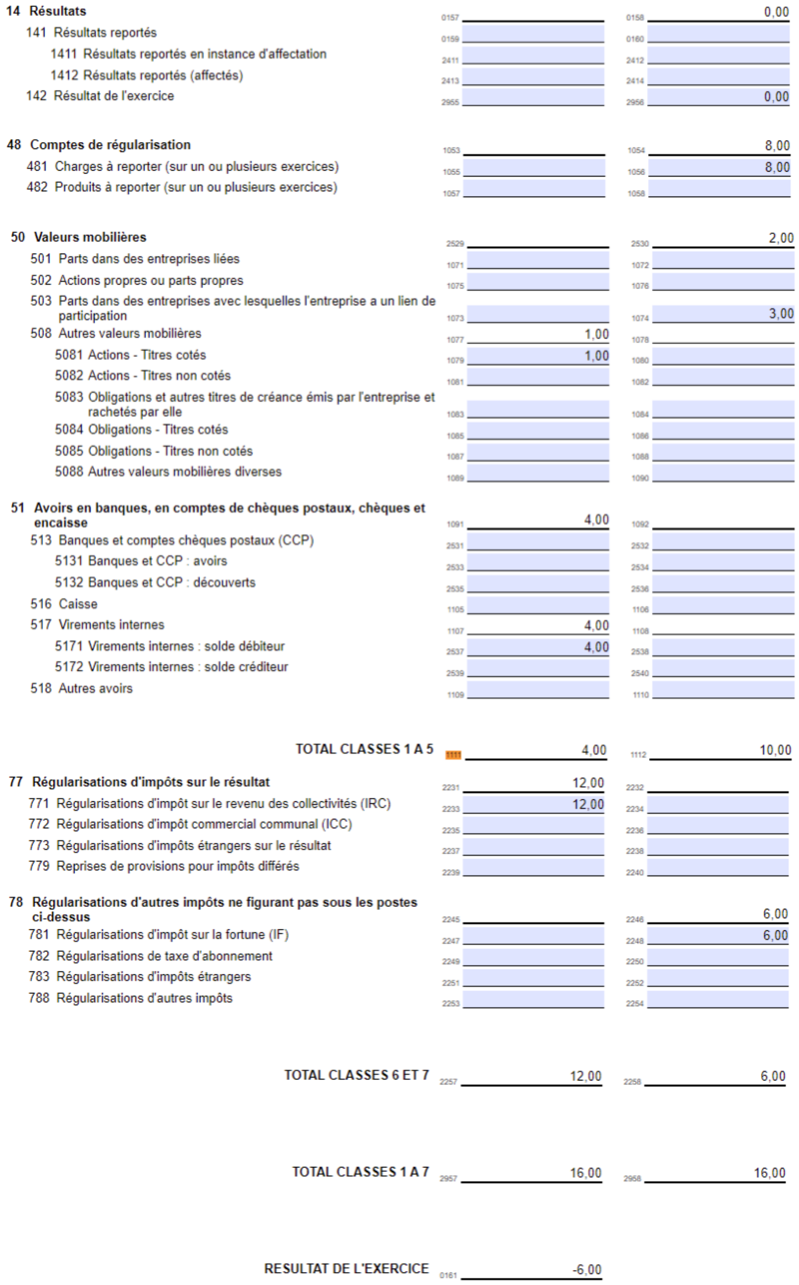
XML 1.1 (until 31.12.2020)	**XML 1.1 (1.1.2021 - 31.03.2021)**	**XML 2.0 (as of 4.1.2021)**

<?xml version="1.0"
<eCDFDeclarations
<FileReference
<eCDFFileVersion>**1.1**
...
<Declarations>
<Declarer>
... AAAAA
 	
<Declaration "CA_BILAN" model=1
    <Year>2019</Year>
    ...
<Declaration "CA_COMPP" model=1
    <Year>2019</Year>
    ...
<Declara. "CA_PLANCOMPTA" model=1
    <Year>2019</Year>
    ...
<Declaration "TVA_..." model=1
    <Year>2019</Year>
    ... model=1
 	
<Declaration "CA_BILAN" model=1
    <Year>2020</Year>
    ...
<Declaration "CA_COMPP" model=1
    <Year>2020</Year>
    ...
<Declara. "CA_PLANCOMPTA" model=1
    <Year>2020</Year>
    ...
 	
<Declaration "TVA_..." model=1
    <Year>2020</Year>
    ...
 	
following years are not accepted
	<?xml version="1.0"
	<eCDFDeclarations
	<FileReference
	<eCDFFileVersion>**1.1**
	...
	<Declarations>
	<Declarer>
	... AAAAA
	
	<Declaration "CA_BILAN" model=1
	    <Year>2019</Year>
	    ...
	<Declaration "CA_COMPP" model=1
	    <Year>2019</Year>
	    ...
	<Declara. "CA_PLANCOMPTA" model=1
	    <Year>2019</Year>
	    ...
	<Declaration TVA_..." model=1
	    <Year>2019</Year>
	    ...
	
	~~<Declaration "CA_BILAN" model=1~~
	~~ <Year>2020</Year>~~
	~~ ...~~
	~~<Declaration "CA_COMPP" model=1~~
	~~ <Year>2020</Year>~~
	~~ ...~~
	~~<Declara. "CA_PLANCOMPTA" model=1~~
	~~ <Year>2020</Year>~~
	~~ ...~~
	
	<Declaration "TVA_..." model=1
	    <Year>2020</Year>
	    ...
	
	following years are not accepted
		<?xml version="1.0"
		<eCDFDeclarations
		<FileReference
		<eCDFFileVersion>**2.0**
		...
		<Declarations>
		<Declarer>
		... AAAAA
		
		<Declaration "CA_BILAN" model=1
		    <Year>2019</Year>
		    ...
		<Declaration "CA_COMPP" model=1
		    <Year>2019</Year>
		    ...
		<Declara. "CA_PLANCOMPTA" model=1
		    <Year>2019</Year>
		    ...
		<Declaration "TVA_..." model=1
		    <Year>2019</Year>
		    ...
		
		<DeclarationGroup>
		  <Declaration "CA_BILAN" model=**2**
		      <Year>2020</Year>
		      ...
		  <Declaration "CA_COMPP" model=**2**
		      <Year>2020</Year>
		      ...
		  <Declara. "CA_PLANCOMPTA" model=**2**
		      <Year>2020</Year>
		      ...
		  <**MappingTable** mapping="standard"/>
		</DeclarationGroup>
		<Declaration "TVA_..." model=1
		    <Year>2020</Year>
		    ...
		
		<DeclarationGroup>
		  <Declaration "CA_BILAN" model=**1**
		      <Year>2021</Year>
		      ...
		   <Declaration "CA_COMPP" model=**1**
		      <Year>2021</Year>
		      ...
		  <Declara. "CA_PLANCOMPTA" model=**1**
		      <Year>2021</Year>
		      ...
		  <**MappingTable** mapping="standard"/>
		</DeclarationGroup>
		<Declaration "TVA_..." model=1
		    <Year>2021</Year>
		    ...
		
		<Declarer>
		... BBBBBB
		<Declaration "TVA_..." model=1
		    <Year>2020</Year>
		    ...
		<Declara. "CA_PLANCOMPTA" model=**2**
		    <Year>2020</Year>
		    ...
		
		<Declaration "TVA_..." model=1
		    <Year>2021</Year>
		    ...
		<Declara. "CA_PLANCOMPTA" model=**1**
		    <Year>2021</Year>
		    ...

3. New features concerning the Standard Chart of Accounts (PCN)

In the course of 2020, it is possible to fill the PCN 2020 in the traditional way, i.e. with values at the aggregated accounts level (i.e. sum fields) without specifying values at the detailed level of ledger accounts (i.e. detail fields).

From 2021 onwards however, the values in the PCN for financial years starting in 2020 or later must be provided at the most detailed level of ledger accounts. Values only in the aggregated accounts are no longer possible, because the mappings defined in the transition table are expressed in terms of ledger accounts.



14 Résultats	0157		0158	0,00
141 Résultats reportés	0159		0160	
1411 Résultats reportés en instance d'affectation	2411		2412	
1412 Résultats reportés (affectés)	2413		2414	
142 Résultat de l'exercice	2955		2956	0,00
48 Comptes de régularisation	1053		1054	8,00
481 Charges à reporter (sur un ou plusieurs exercices)	1055		1056	8,00
482 Produits à reporter (sur un ou plusieurs exercices)	1057		1058	
50 Valeurs mobilières	2529		2530	2,00
501 Parts dans des entreprises liées	1071		1072	
502 Actions propres ou parts propres	1075		1076	
503 Parts dans des entreprises avec lesquelles l'entreprise a un lien de participation	1073		1074	3,00
508 Autres valeurs mobilières	1077	1,00	1078	
5081 Actions - Titres cotés	1079	1,00	1080	
5082 Actions - Titres non cotés	1081		1082	
5083 Obligations et autres titres de créance émis par l'entreprise et rachetés par elle	1083		1084	
5084 Obligations - Titres cotés	1085		1086	
5085 Obligations - Titres non cotés	1087		1088	
5088 Autres valeurs mobilières diverses	1089		1090	
51 Avoirs en banques, en comptes de chèques postaux, chèques et encaisse	1091	4,00	1092	
513 Banques et comptes chèques postaux (CCP)	2531		2532	
5131 Banques et CCP : avoirs	2533		2534	
5132 Banques et CCP : découverts	2535		2536	
516 Caisse	1105		1106	
517 Virements internes	1107	4,00	1108	
5171 Virements internes : solde débiteur	2537	4,00	2538	
5172 Virements internes : solde créditeur	2539		2540	
518 Autres avoirs	1109		1110	
TOTAL CLASSES 1 A 5	1111	4,00	1112	10,00
77 Régularisations d'impôts sur le résultat	2231	12,00	2232	
771 Régularisations d'impôt sur le revenu des collectivités (IRC)	2233	12,00	2234	
772 Régularisations d'impôt commercial communal (ICC)	2235		2236	
773 Régularisations d'impôts étrangers sur le résultat	2237		2238	
779 Reprises de provisions pour impôts différés	2239		2240	
78 Régularisations d'autres impôts ne figurant pas sous les postes ci-dessus	2245		2246	6,00
781 Régularisations d'impôt sur la fortune (IF)	2247		2248	6,00
782 Régularisations de taxe d'abonnement	2249		2250	
783 Régularisations d'impôts étrangers	2251		2252	
788 Régularisations d'autres impôts	2253		2254	
TOTAL CLASSES 6 ET 7	2257	12,00	2258	6,00
TOTAL CLASSES 1 A 7	2957	16,00	2958	16,00
RESULTAT DE L'EXERCICE	0161	-6,00		

Note that account 142 “Result for the financial year” is always as in the example above, i.e. with field 2955 empty and field 2956 at zero. This is because the chart of accounts (PCN) represents the situation after the closure and the amount in account 142 (i.e. the balance of charges and income) has already been carried forward, so it is empty again.

4. XML file structure version 2.0

In the current document, if not specified differently, XML file means XML file corresponding to the eCDF file version 2.0.

4.1. Well-formed XML

A file of financial data to be transferred into eCDF has to consist of well-formed XML. This means that it has to meet the syntactic constraints set by the XML standard. It's also pointed out that some predefined entities (special characters) and also the use of CDATA are not allowed as values in eCDF XML files. When such characters are present as the value of an element (alphanumeric field), then their encoded representation has to be used:

Left angle bracket	<	&lt;
Right angle bracket	>	&gt;
Ampersand	&	&amp;

4.2. Validity against the XML schema

The second prerequisite of any XML file for eCDF is its validity against the eCDF XML schema. This schema corresponding to the present file version is available for download in the interface developers' area in the menu « Developers / Documentation ».

4.3. Elements of the XML file for eCDF

The XML elements as prescribed by the eCDF XML schema are explained below. A small XML sample file can be found in section 4.5. Other more complete examples are outside of this document in the menu « Developers / Documentation».

Please note that empty elements having no value cannot be included in the file. This means that elements like the following ones are explicitly prohibited:

```
<TextField id="02"/>

<TextField id="02"></TextField>
```

4.3.1. XML header

Any XML file for eCDF has to start with the following tag that indicates the XML version and character encoding (UTF-8):

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

This element must not be modified. The version attribute contained in this element must always be 1.0 and must not be confused with the element "ECDFFileVersion".

The XML file for eCDF has to be encoded in UTF-8. It's not sufficient to just specify UTF-8 in the XML header but the eCDF XML file really has to be encoded and saved as UTF-8, otherwise it will be rejected by eCDF.

4.3.2. Document Root : eCDFDeclarations

The mandatory element « eCDFDeclarations » contains all the required elements of an eCDF XML file. It has to specify the *xmlns* attribute indicating the eCDF namespace.

```
<eCDFDeclarations xmlns="http://www.ctie.etat.lu/2011/ecdf">
```

The *xmlns* attribute is a predefined text that must not be changed. It is not a real URL accessible by Internet and the year contained in the text is not related in any way to the reference years of the declarations in the file.

4.3.3. FileReference element

The value of the « FileReference » element is the file reference which allows to uniquely identify each XML file transferred into eCDF.

```
<FileReference>000000XaaaammjjThhmmssNN</FileReference>
```

The reference must be unique and respect the format below:

```
000000XaaaammjjThhmmssNN
```

Position 1 - 6:	eCDF prefix of the agent (company using the XML interface) (alphanumeric characters)
Position 7:	file type (X for XML files)
Position 8 - 15:	valid creation date of your file in the format yyyymmdd
Position 16:	the « T » character (Time)
Position 17 - 22:	valid creation time of your file in the format hhmmss
Position 23 - 24:	sequence number (NN) between 01 and 99 to ensure the uniqueness of your file names created in the same second

The eCDF prefix of the agent company should be configurable in the developed interface as it is unique for every company using the interface. This prefix is generated and displayed when the agent company using the interface accesses the menu « File Transfer » of the eCDF website for the first time and it remains visible afterwards by following the « unique reference » link available on the « File Transfer » page.

The « File transfer » menu is only available if the user has previously submitted an XML request form «Request for use of the eCDF system » for an authorized interface, which has been approved by the CTIE.

In addition, the physical name of the XML file on the disk of the interface user has to match the reference file completed with the file extension « .xml ».

Example: FileReference: « 000000X20200711T15314401 »
Physical name: « 000000X20200711T15314401.xml »

For the tests in the interface developers' area and only for them, any value that respects the prefix length is accepted, including the prefix « 000000 ».

4.3.4. eCDFFileVersion element

The « eCDFFileVersion » specifies the eCDF file version (i.e. the specification) to which the XML corresponds to and according to which it will be validated.

```
<eCDFFileVersion>2.0</eCDFFileVersion>
```

The detailed documentation of this file version can be found in the eCDF menu “Developer / Documentation” and this documentation also lists the different form types, models and reference years that can be transferred using this file version.

4.3.5. Interface element

The « Interface » element indicates the unique identifier of the XML interface used to generate this file.

```
<Interface>IIIII</Interface>
```

The value of this element is the identifier as it has been assigned to the interface by eCDF after the validation of the first developer access request for the interface. This identifier has been sent to the developer in the developer access request validation email. The identifier is also displayed in the menu «Developers / Interface management» of the area dedicated to developers.

The interface identifier is fixed and must be present in each file generated by the interface, even if the interface is used by different companies. The presence of a valid interface identifier is one of the conditions for acceptance of XML files by the eCDF. This also applies to test files and the example identifier «IIIII» must be replaced by the assigned interface identifier since the first generation of an XML file by the interface.

4.3.6. Agent element

The « Agent » element contains information about the agent company that prepares and transfers the XML file, that is to say the company using your interface.

This company is identified by the three following identifiers: national id number, « Registre du Commerce et des Sociétés » number (RCS No.), VAT number. The three corresponding elements provided for these identifiers must always be present and filled in the XML file.

As these three identifiers and the eCDF file prefix (cf. chapter 4.3.3) are unique to each company using the interface, they must be configurable in the configuration page in the developed interface, which must contain the following four parameters:

- eCDF prefix
- national id number (alias matricule)
- RCS number (i.e. the company register number)
- VAT number

These settings specific to the agent must remain fixed and stable over time and in the configuration screens of the XML interface, they must be clearly separated from the declarer’s settings, in order to avoid any unintended change in the agent setting.

In case of incorrect settings, eCDF reports an error related to the "agent identification numbers". It is therefore advisable to use the term "Agent" in the configuration screen of the XML interface to better guide the user in finding the problem.

During the development phase, when the interface developer tests the structure of files generated by his interface using the test feature available in the interface developers' area, these user identifiers are part of the information that is not compared against values actually existing in the databases of the State. Only a simple length control is done.

```
<Agent>
      <MatrNbr>11111111111</MatrNbr>
      <RCSNbr>B111111</RCSNbr>
      <VATNbr>11111111</VATNbr>
</Agent>
```

However, during the official transfer of real financial data to the eCDF system by a company using the interface, the identifiers of this company will be checked. They must exactly match the identifiers listed in the access request to the system in the form « Request for use of the eCDF system ».

If the company does not have a RCS number or a VAT number, the corresponding element must contain the value « NE » (not available). (See sections 4.3.6.2 and 4.3.6.3)

The identification numbers must not contain spaces or other separators.

The fourth element required for identifying the agent is the "eCDF prefix" already described in chapter 4.3.3.

4.3.6.1. MatrNbr element

The value of the « MatrNbr » element is the national id number of the agent company, which always exists.

During the official transfer, its validity is checked as well as its consistency with the two following numbers, namely the RCS number and the VAT number.

```
<MatrNbr>11111111111</MatrNbr>       (11 position format)
<MatrNbr>1111111111111</MatrNbr>     (13 position format)
```

For the tests in the developers' area, it's only checked that this element has a length of 11 or 13 positions.

4.3.6.2. RCSNbr element

The value of the « RCSNbr » element is the « Registre de Commerce et des Sociétés » number (RCS No.) of the agent company transferring the file.

In most cases, companies have such a number, which must then be indicated.

```
<RCSNbr>B111111</RCSNbr>
```

The first letter may be different from B but has to be uppercase. This first letter cannot be followed immediately by one or more spaces or zeros.

However, if the company did not get such a number from the « Registre de Commerce et des Sociétés » (RCS No.), then the corresponding element must contain the default value « NE » (not available).

```
<RCSNbr>NE</RCSNbr>
```

For the tests in the developers' area, the value of this element must have a maximum length of 7 or be « NE ».

4.3.6.3. VATNbr element

The value of the « VATNbr » element is the VAT number of the agent company transferring the file.

In most cases, companies have such a number, which must then be indicated. The VAT number must be specified without the "LU" letters.

```
<VATNbr>11111111</VATNbr>
```

However, if the company does not have such a VAT number, then the corresponding element must contain the default value « NE » (not available).

```
<VATNbr>NE</VATNbr>
```

For the tests in the developers' area, the value of this element must have a length of 8 or be « NE ».

4.3.7. Declarations element

After the information about the file, the interface used to generate the file and the agent company that transfers the file, the « Declarations » element contains the financial data. These declarations must be structured exactly as follows.

The « Declarations » element must contain one or more « Declarer » elements which group the declarations corresponding to the same declarer.

4.3.8. Declarer element

The « Declarer » element groups all information corresponding to the same declarer.

A declarer is identified by the three following identifiers: national id number, « Registre de Commerce et des Sociétés » number (RCS No.), VAT number. The three elements provided for these identifiers must always be present and filled.

During the development phase, if the interface developer tests the structure of files generated by his interface using the test feature available in the interface developers' area, these declarer identifiers are part of the information that is not compared to the real values existing in the State's databases and there is no control on the format.

```
<Declarer>
      <MatrNbr>22222222222</MatrNbr>
      <RCSNbr>B222222</RCSNbr>
      <VATNbr>22222222</VATNbr>
      <Declaration language="FR" type="CA_PLANCOMPTA" model="1">
            <Year>2019</Year>
            <Period>1</Period>
            ...
      <DeclarationGroup>
            <Declaration language="FR" type="CA_PLANCOMPTA" model="2">
                  <Year>2020</Year>
                  <Period>1</Period>
                  ...
```

During the official transfer of real financial data to the eCDF system, the identifiers of this company will be validated. They must exactly match the identifiers specified by the agent company when adding he declarer to his eCDF account.

The « Declarer » element may only be present if it contains at least one declaration and all declarations for the same declarant bust be grouped into the same « Declarer » element. This means that an XML file may only contain one single « Declarer » element for each declarer.

Depending on the type and reference year of the declaration, declarations must be added into the « Declarer » element either as:

- individual declaration represented by a « Declaration » element

  This applies to all VAT declarations and to annual accounts with financial years starting years earlier than 2020, which need to be put into the « Declarer » element as individual « Declaration » elements. This also applies to the exceptional case of annual accounts filed for financial years beginning in 2020 or later, consisting solely of the standard chart of accounts (PCN).

  This is the classic behaviour as it used to be in the XML 1.1 file version for eCDF.

- group of declarations grouped by a « DeclarationGroup » element

This applies to all annual accounts deposits for financial years starting in 2020 or later, consisting of more than one element. For these annual accounts deposits all its elements (« Declaration »; « MappingTable ») must be grouped inside of the same « DeclarationGroup » element, which is located right inside the « Declarer » element.

This is a new behaviour introduced in the XML 2.0 file version for eCDF. More information is available in chapter 4.3.13.

4.3.8.1. MatrNbr element

The value of the « MatrNbr » element is the national id number of the declarer. It always exists.

```
<MatrNbr>22222222222</MatrNbr>          (11 position format)
<MatrNbr>2222222222222</MatrNbr>        (13 position format)
```

4.3.8.2. RCSNbr element

The value of the « RCSNbr » element is the « Registre de Commerce et des Sociétés » number (RCS No.) of the declarer.

In most cases, companies have such a number, which must then be indicated. The number consists of one uppercase letter and between 2 and 6 figures. The first figure cannot be a 0.

```
<RCSNbr>B222222</RCSNbr>
```

However, if the declarer did not get such a number from the « Registre de Commerce et des Sociétés » (RCS No.), then the corresponding element must contain the default value « NE » (not available).

```
<RCSNbr>NE</RCSNbr>
```

4.3.8.3. VATNbr element

The value of the « VATNbr » element is the VAT number of the declarer.

In most cases, companies have such a number, which must then be indicated. The VAT number must be specified without the “LU” letters.

```
<VATNbr>22222222</VATNbr>
```

However, if the declarer does not have such a VAT number, then the corresponding element must contain the default value « NE » (not available).

```
<VATNbr>NE</VATNbr>
```

4.3.9. Declaration element

The « Declaration » element is a declaration of the declarer and is identified by the « type » and « model » attributes and the « Year » and « Period » elements. The same declaration can be present only once per declarer, either immediately under « Declarer » or inside of a « DeclarationGroup ».

```
<Declaration type="CA_BILAN" model="1" language="FR">
      <Year>2019</Year>
      <Period>1</Period>
      <FormData>
      ...
```

The « Declaration » element must specify all three attributes « type », « model » and « language ».

The « type » attribute indicates the type of declaration in the « Declaration » element in question. The possible values are listed in the following table:

« type » attribute	Type
`CA_BILAN`	Annual accounts - Balance sheet
`CA_BILANABR`	Annual accounts - Abridged balance sheet
`CA_COMPP`	Annual accounts - Profit and loss account
`CA_COMPPABR`	Annual accounts - Abridged profit and loss account
`CA_PLANCOMPTA`	Annual accounts - Chart of accounts
`TVA_DECM`	VAT - Monthly declaration
`TVA_DECT`	VAT - Quarterly declaration
`TVA_DECA`	VAT - Yearly declaration
`TVA_DECAS`	VAT - Yearly simplified declaration
`TVA_LICM`	VAT - Monthly EC sales list (deliveries)
`TVA_LICT`	VAT - Quarterly EC sales list (deliveries)
`TVA_PSIM`	VAT - Monthly EC sales list (services)
`TVA_PSIT`	VAT - Quarterly EC sales list (services)
`AN_TABACAM`	Appendices – Tables of acquisitions / amortisable expenditures

The « model » attribute specifies to which form model the declaration corresponds. The different types of declarations, models and reference years allowed for this file version are listed in the menu “Developers / Documentation”. If for a given reference year additional models get available, then the value to be specified in the “model” attribute will be given to the developers together with all other information concerning these forms.

The « language » attribute for each statement indicates the language to be considered as the eCDF deposit language. A PDF form in this language will be generated by default by eCDF to display the transferred data.

Possible values are « FR » for French, « DE » for German and « EN » for English.

A maximum of 999 declarations are allowed per XML file.

4.3.10. Year element

The « Year » element contains the reference year of the declaration.

For annual accounts declarations, the reference year is the calendar year of the start date of the financial year of the declaration. For a declaration with a financial year from March 15th 2012 to March 14th 2013, the reference year is 2012.

```
<Year>2012</Year>
```

For the VAT declarations, the reference year corresponds to the calendar year of the declaration period.

4.3.11. Period element

The « Period » element indicates the period of the year to which the declaration refers.

For annual accounts declarations, which have by definition always a yearly period, this value is always « 1 ».

```
<Period>1</Period>
```

For VAT declarations, the « Period » element may also take other values since besides those declarations that have to be deposited by year (« Period » of 1), others have to be deposited monthly (« Period » from 1 to 12) or quarterly (« Period » from 1 to 4).

4.3.12. FormData element

The « FormData » element contains all the fields of the declaration.

Each field appears below « FormData » as an individual element. The element's type has to correspond to the specified field type.

The three following types of field elements exist:

`TextField` A text field is represented by a « TextField » element. It is used for comment fields (i.e. field 2385 of the chart of accounts), reference fields, dates and currencies.

`NumericField` A numeric field is represented by a « NumericField » element

`Choice` A boolean field is represented by a « Choice » element
The value « 0 » means not checked (false)
The value « 1 » means checked (true)

In addition to the fields that are freely distributed on the declaration and which have each their own field identifier, there are also fields that are formatted as a table. These fields can be identified by the « id » identifier of the table column and the « num » identifier of the table row in which the field is located.

```
<FormData>
      <TextField id="01">01/01/2012</TextField>
      <TextField id="02">31/12/2012</TextField>
      <TextField id="03">EUR</TextField>
      <NumericField id="0101">500</NumericField>
      <NumericField id="0102">217851,2</NumericField>
      <NumericField id="0103">24992,08</NumericField>
      <Choice id="9991">1</Choice>
      <Choice id="9992">0</Choice>
      <Table>
            <Line num="1">
                  <TextField id="411">descr1</TextField>
                  <NumericField id="412">30,00</NumericField>
                  <NumericField id="413">100,00</NumericField>
            </Line>
            <Line num="2">
                  <TextField id="411">descr2</TextField>
                  <NumericField id="412">60,00</NumericField>
                  <NumericField id="413">200,00</NumericField>
            </Line>
      </Table>
      <Table>
            <Line num="1">
                  <TextField id="501">descr1</TextField>
                  <TextField id="502">descr2</TextField>
```

The validation rules document available under the « Forms » menu in the « Validation Rules » tab of the form contains a column which specifies for every field the type of value required. For numeric fields, this column can specify that the value has to be either integer, decimal or positive but in any case, it's a numeric value and the «NumericField» element has to be used in the XML. The other fields are either text («TextField» element) or check boxes («Choice» element).

It is imperative that the developed interface exports the declaration fields as the correct element type; otherwise the XML file will be rejected due to a structure error.

It is not allowed to insert fields under «FormData» with identifiers that do not exist for this type of declaration.

4.3.12.1. Id attribute

Each element corresponding to a field must contain an « id » attribute, which indicates the identifier of the field. This field identifier is indicated next to the fields on the forms that are available in PDF format in the « Forms » menu of the eCDF website.

In everyday language, this identifier is often called « field number ». However, this is not a numeric identifier but a string identifying the field. For every « TextField », « NumericField » or « Choice » element representing a field, this identifier has to be entered in the « id » attribute in exactly the same way as printed on the form.

In the « id » attribute, the value « 352 » is actually distinct from the value « 0352 ».

For fields being part of a table, the « id » is identical for all the fields of the same column.

4.3.12.2. NumericField element

If the « Validation Rules » tab indicates that a field is numeric, then the value of this field has to be included in the XML file as a « NumericField » element.

The value of a « NumericField » can be an integer or a decimal. The only separator accepted for decimals is the comma (« , »). The point (« . ») is not accepted, neither as decimal nor thousands separator.

The number of decimals allowed is zero, one or two.

```
<NumericField id="0101">500</NumericField>
<NumericField id="0102">217851,2</NumericField>
<NumericField id="0103">24992,08</NumericField>
```

It is important to note that eCDF does not consider in the same way a « NumericField » element with value «0» and a « NumericField » element not present in the XML file. A « NumericField » element with value « 0 » is a correct value while a missing element is not.

So if for two fields, a business rule specifies that only one of them can have a value, only the element corresponding to the field containing the value may be present in the XML file. In this case the second field must not be present, not even with the value « 0 ».

In case of negative values, no space is accepted between the negative sign and the numbers.

4.3.12.3. TextField element

If the « Validation Rules » states that a field must contain an alphanumeric value, then the value of this field has to be included in the XML file as a « TextField » element. Date and currency fields of the forms are also considered as text fields.

If the « TextField » element is used for dates, then its formatting has to be the one specified in the « Validation Rules » tab for the type of declaration in question.

```
<TextField id="02">31/12/2012</TextField>
```

If the « TextField » element is used to indicate a currency, this must be done by using the currency code with three capital letters as specified by ISO 4217.

```
<TextField id="03">EUR</TextField>
```

The schema limits elements of this type to 1800 characters. This is also the maximum length if no more specific rule specifies a shorter length (see the validation rules document for more information). As already explained in section 2.1, the XML standard requires that for some characters, their encoded representation has to be used.

```
<TextField id="2385">&apos;X&apos;&lt;&gt;&quot;Y&quot;</TextField>
```

4.3.12.4. Choice element

The only two values for the « Choice » element are « 0 », meaning « unchecked » (false), and « 1 » which means « checked» (true).

```
<Choice id="9991">0</Choice>
<Choice id="9992">1</Choice>
```

4.3.12.5. Table element

Fields grouped in a table are identified by the column identifier « id » and the line identifier « num » of the field.

The « Table » element groups all the lines of one table, with each line being represented by a « Line » element.

```
<Table>
      <Line num="1">
            <TextField id="411">descr1</TextField>
            <NumericField id="412">30,00</NumericField>
            <NumericField id="413">100,00</NumericField>
      </Line>
      <Line num="2">
            <TextField id="411">descr2</TextField>
            <NumericField id="412">60,00</NumericField>
            <NumericField id="413">200,00</NumericField>
      </Line>
</Table>
```

The « Table » elements must always be at the end of the « FormData » element, right behind the fields, wherever they are located on the declaration form. If this order is not respected, the XML file will be rejected.

The number of rows allowed in a table is determined by the form’s visual representation (i.e. layout).

- If the table is followed by a page for additional rows (e.g. in the "Recapitulative statements" and "Tables of acquisitions/ amortisable expenditures" forms), then the number of rows in the table is limited only by the maximum size of the resulting XML file.
- In all other cases, when no specific page for additional rows is available, the number of rows in the XML file is limited to the number of rows visible on the form (e.g. in the annual VAT return, the "Appendix to Operational expenditures" table is therefore limited to 42 rows).

4.3.12.6. Line element

The « Line » element groups all the fields of a table line and the « num » attribute of this element is a running number which specifies the line’s position in the table. The fields contained in the « Line » element may be of the three types explained above: NumericField, TextField and Choice.

The « Line » elements have to put into ascending order of their « num » attribute, which is a positive integer. The « num » attribute has to be unique by « Table » element.

4.3.13. DeclarationGroup element

In annual accounts deposits for financial years starting in 2020 or later and consisting of several elements (« Declaration »; « MappingTable »), all these elements must be grouped inside of the same «DeclarationGroup» element, which is located right inside the « Declarer » element.

```
<Declarer>
      ...
      <DeclarationGroup>
            <Declaration language="FR" type="CA_PLANCOMPTA" model="2">
                  <Year>2020</Year>
                  <Period>1</Period>
                  ...
                  <Choice id="2940">1</Choice>
                  <Choice id="2941">1</Choice>
                  <Choice id="2942">1</Choice>
            <Declaration language="FR" type="CA_BILAN" model="2">
                  <Year>2020</Year>
                  <Period>1</Period>
                  ...
            <Declaration language="FR" type="CA_COMPP" model="2">
                  <Year>2020</Year>
                  <Period>1</Period>
                  ...
            <MappingTable mapping="standard"/>
            ...
      </DeclarationGroup>
```

Dedicated “choice” fields in the “Standard chart of account” (e.g. 2940, …) specify what other documents are present in the « DeclarationGroup ».

An annual accounts deposit for financial years starting in 2020 or later does usually consist of the following four elements which must be grouped inside of the « DeclarationGroup » element:

- the “Standard chart of account”
- the “Balance sheet” or “Abridged Balance Sheet”
- the “Profit and loss account” or “Abridged profit and loss account”
- the “Mapping table”

The exception to this is the case explained in chapter 2.2.2 of companies for which the “Non-automated generation of the balance sheet and the profit and loss account” option on the standard chart of accounts (PCN) is allowed. These annual accounts filings do only consist of one single standard chart of accounts (PCN) which must not be inside of a « DeclarationGroup » but immediately under « Declarer ».

```
<Declarer>
      ...
      <Declaration language="FR" type="CA_PLANCOMPTA" model="2">
            <Year>2020</Year>
            <Period>1</Period>
            ...
            <Choice id="2952">1</Choice>
</Declarer>
```

4.3.14. MappingTable element

Every annual accounts deposit for financial years starting in 2020 or later, which contains an complete or abridged balance sheet and a complete or abridged profit and loss account must also contain a « MappingTable » element, which indicates according to which calculation rules, the values in the balance sheet and profit and loss account.have been calculated based upon the values from the standard chart of accounts.

The « MappingTable » must always be the last element inside the « DeclarationGroup ». Otherwise the XML file will be rejected.

4.3.14.1. “standard” mapping table

The “Commission des normes comptables (CNC)” has provided a standard mapping table which can be found in eCDF under the menu “Forms / Mapping Tables”. It’s also available for download as XML.

If in an XML file, the values inside both the (abridged) balance sheet and the (abridged) profit and loss account have been calculated according to the standard mapping table, then it’s sufficient to specify this inside of the « DeclarationGroup » by adding the following element:

```
<MappingTable mapping="standard"/>
```

4.3.14.2. “custom” mapping table

As soon as in the XML file one value in either the (abridged) balance sheet or in the (abridged) profit and loss account has been calculated differently than specified in the standard mapping table, this custom calculation method for this declaration needs to be specified inside of the « DeclarationGroup » by adding a « custom » « MappingTable».

The « custom » type « MappingTable » must contain two « TargeForm » elements. One for each (abridged) balance sheet or (abridged) profit and loss account present within the « DeclarationGroup ». There must be no « TargeForm » for a form type not present in the « DeclarationGroup ».

In a «MappingTable» of the "Custom" type, at least one of the « TargeForm » elements must actually contain custom calculations, different from those of the standard mapping table.

```
<MappingTable mapping="custom">
      <TargetForm type="CA_BILAN" year="2020" model="2" >
            <Mapping fieldId="137">
                  <PcnAccount debitId="0437" creditId="0438"/>
                  <PcnAccount debitId="0441" creditId="0442"/>
            </Mapping>
            <Mapping fieldId="141"/>
            <Mapping fieldId="335">
                  <PcnAccount debitId="0185" creditId="0186"/>
            </Mapping>
            <Mapping fieldId="337">
                  <PcnAccount debitId="0195" creditId="0196"/>
                  <PcnAccount debitId="0199" creditId="0200"/>
                  <PcnAccount debitId="0201" creditId="0202"/>
            </Mapping>
        </TargetForm>
        <TargetForm type="CA_COMPP" year="2020" model="2">
            <Mapping fieldId="653"/>
            <Mapping fieldId="655">
                <PcnAccount debitId="2627" creditId="2628"/>
                <PcnAccount debitId="2629" creditId="2630"/>
            </Mapping>
        </TargetForm>
    </MappingTable>
```

example : calculation method for fields in balance sheet and profit and loss account

The previous « MappingTable » specifies the following differences compared to the standard mapping table. This example only shows the technical XML aspects but is not necessarily a valid mapping from a business perspective.

Balance S. - Assets [1]	field 137	custom calculation : (0437-0438)+(0441-0442)
	field 141	custom calculation : no calculation anymore
Capital, … [2]	field 335	custom calculation : (0186-0185)
	field 337	custom calculation : (0196-0195)+(0200-0199)+(0202-0201)
Profit and loss accounts [3]	field 653	custom calculation : no calculation anymore
	field 655	custom calculation : (2628-2627)+(2630-2629)

```
    <MappingTable mapping="custom">
        <TargetForm type="CA_BILAN" year="2020" model="2" >
            <Mapping fieldId="137">
                <PcnAccount debitId="0437" creditId="0438"/>
                <PcnAccount debitId="0441" creditId="0442"/>
            </Mapping>
            <Mapping fieldId="141"/>
        </TargetForm>
        <TargetForm type="CA_COMPP" year="2020" model="2"/>
    </MappingTable>
```

exemple : mapping bilan custom et mapping cpp standard

If the "Custom" mapping only concerns one of the (abridged) balance sheet items or the (abridged) profit and loss account, then the other one that uses the standard mapping must be contained in the « MappingTable » as an empty « TargetForm ».

4.3.15. TargetForm element

Inside the « MappingTable » element of the "custom" type, there must be two « TargetForm » elements. One for each (abridged) balance sheet and (abridged) profit and loss account present in the « DeclarationGroup». There must be no « TargetForm » for a form type not present in the « DeclarationGroup».

[1] Balance sheets - Assets : solde débiteur - solde créditeur
[2] Balance sheets - Capital, reserves and liabilities : solde créditeur - solde débiteur
[3] Profit and loss accounts : solde créditeur - solde débiteur

```
<TargetForm type="CA_BILAN" year="2020" model="2">
            <Mapping fieldId="137">
                  ...
      </TargetForm>
```

The « TargetForm » element lists for the specified form type, the custom calculations, different from those in the standard mapping table. Such a custom calculation is represented by the « Mapping » element. The « TargetForm » element must not contain and repeat standard calculations, already specified in the standard mapping table.

If in the "Custom" type « MappingTable », one of the two referenced forms has been calculated according to the standard mapping table, then it must be represented by an empty « TargetForm » element.

```
TargetForm type="CA_BILAN" year="2020" model="2"/>
```

4.3.16. PcnAccount element

The « PcnAccount » represents one account from the standards chart of accounts (PCN), corresponding to one line on the PCN), and contains as attributes, the field numbers of the line's debit ("Net debit balance") and credit ("Net credit balance") fields. The order of the « PcnAccount » elements is of no importance.

```
<PcnAccount debitId="0437" creditId="0438"/>
```

4.3.17. Mapping Element

A « Mapping » describes for a specific field, indicated by the attribute, how it is calculated based upon the field values from the standard chart of accounts (PCN).

Each custom calculation, different from those in the standard mapping table, must be indicated inside the « TargetForm » element using a « Mapping » element.

This is also the case for the fields having lost their calculation. If according to the standard mapping table, a field in the (abridged) balance sheet items or the (abridged) profit and loss account used to be filled with values coming from the standard chart of accounts (PCN), but the custom mapping now assigns these values to different fields, this may result in a field getting assigned no values anymore. In this case, the field having lost its calculation must be specified inside the « custom » mapping table using an empty « Mapping » element.

It is not allowed to add and repeat in « TargetForm » a « Mapping » that already exists in the standard mapping table and such a redundant calculation will invalidate the XML file.

```
<Mapping fieldId="137">
      <PcnAccount debitId="0437" creditId="0438"/>
      <PcnAccount debitId="0441" creditId="0442"/>
</Mapping>
<Mapping fieldId="141"/>
```

example : field 141 does not get any values from the PCN anymore

The « Mapping » elements groups all « PcnAccount » elements intervening in the calculation of the field specified by the « fieldId » attribute.

The way in which the debit and credit fields are used in the calculation (subtracted or added) is automatically determined by the type of destination field (balance sheet assets, balance sheet liabilities, profit and loss account). (see example in chapter 4.3.14.2.)

4.4. Size of the generated XML

An XML file to be transferred into eCDF may contain declarations on behalf of one or more declarers. All declarations regarding a declarer have to be grouped in the same « Declarer » element.

eCDF has been designed to accept large files. However, it is recommended to the interface users to generate files of reasonable size.

No blank characters should be added to the file for the sole purpose of formatting, because this unnecessarily blows up the file size. It’s however recommended to add carriage returns for each line for a better readability.

4.5. XML file structure example

The following example is not exhaustive in terms of represented fields but sufficient to illustrate the structure of an XML file with data for multiple declarers, each with several declarations of different types.

Apart from the value of the « FileReference» element, the file header is the same for all files generated by a user of the interface.

```
<?xml version="1.0" encoding="UTF-8"?>
<eCDFDeclarations xmlns="http://www.ctie.etat.lu/2011/ecdf">
    <FileReference>000000X20130711T15314401</FileReference>
    <eCDFFileVersion>1.1</eCDFFileVersion>
    <Interface>IIIII</Interface>
    <Agent>
        <MatrNbr>11111111111</MatrNbr>
        <RCSNbr>B111111</RCSNbr>
        <VATNbr>11111111</VATNbr>
    </Agent>
```

Below the file header, the declarations are inside the « Declarations » element in which they are grouped by declarer.

The data block starts with the « Declarations » element. This block contains a « Declarer » element corresponding to a first declarer with his three declarations.

```
    <Declarations>
        <Declarer>
            <MatrNbr>22222222222</MatrNbr>
            <RCSNbr>B222222</RCSNbr>
            <VATNbr>22222222</VATNbr>
            <Declaration type="CA_BILANABR" model="2" language="FR">
                <Year>2012</Year>
                <Period>1</Period>
                <FormData>
                    <TextField id="01">01/01/2012</TextField>
                    <TextField id="02">31/12/2012</TextField>
                    <TextField id="03">EUR</TextField>
                    <NumericField id="151">91745,77</NumericField>
                    <TextField id="1151">ABC.&lt;DEF.</TextField>
                    <NumericField id="152">72638,40</NumericField>
                    <TextField id="1152">Ref.A.1</TextField>
                    ...
                    ...
                    <NumericField id="330">6844,71</NumericField>
                    <NumericField id="405">123522,90</NumericField>
                    <NumericField id="406">108827,06</NumericField>
                </FormData>
            </Declaration>
            <Declaration type="CA_COMPPABR" model="2" language="FR">
                <Year>2012</Year>
                <Period>1</Period>
                <FormData>
                    <TextField id="01">01/01/2012</TextField>
                    <TextField id="02">31/12/2012</TextField>
                    <TextField id="03">EUR</TextField>
                    <NumericField id="639">54606,08</NumericField>
                    <NumericField id="640">37499,59</NumericField>
                    <TextField id="1640">Ref.B.2</TextField>
                    ...
                    ...
                    <NumericField id="737">54606,08</NumericField>
                    <NumericField id="738">37499,59</NumericField>
                </FormData>
            </Declaration>
            <Declaration type="CA_PLANCOMPTA" model="1" language="FR">
                <Year>2012</Year>
                <Period>1</Period>
                <FormData>
```

```
                    <TextField id="01">01/01/2012</TextField>
                    <TextField id="02">31/12/2012</TextField>
                    <TextField id="03">EUR</TextField>
                    <NumericField id="0102">2425,48</NumericField>
                    <NumericField id="0106">49307,23</NumericField>
                    <NumericField id="0117">46881,75</NumericField>
                    <NumericField id="0132">12275,51</NumericField>
                    ...
                    ...
                    <NumericField id="2347">27537,04</NumericField>
                    <NumericField id="2353">27537,04</NumericField>
                    <TextField id="2385">by Max &amp; Moritz</TextField>
                </FormData>
            </Declaration>
        </Declarer>
```

A second « Declarer » element contains data for the second declarer for whom one VAT declaration and two annual accounts declarations are contained in the file.

```
        <Declarer>
            <MatrNbr>3333333333333</MatrNbr>
            <RCSNbr>B333333</RCSNbr>
            <VATNbr>33333333</VATNbr>
            <Declaration type="TVA_DECA" model="1" language="FR">
                <Year>2015</Year>
                <Period>1</Period>
                <FormData>
                    <NumericField id="233">1</NumericField>
                    <NumericField id="234">1</NumericField>
                    <NumericField id="235">31</NumericField>
                    <NumericField id="236">12</NumericField>
                    <NumericField id="012">5010,00</NumericField>
                    <Choice id="204">1</Choice>
                    <Choice id="205">0</Choice>
                    ...
                    ...
                    <NumericField id="414">90,00</NumericField>
                    <NumericField id="415">300,00</NumericField>
                    <Table>
                      <Line num="1">
                              <TextField id="411">descr1</TextField>
                              <NumericField id="412">30,00</NumericField>
                              <NumericField id="413">100,00</NumericField>
                      </Line>
                      <Line num="2">
                              <TextField id="411">descr2</TextField>
                              <NumericField id="412">60,00</NumericField>
                              <NumericField id="413">200,00</NumericField>
                      </Line>
                    </Table>
                </FormData>
            </Declaration>
            <Declaration type="CA_BILAN" model="1" language="EN">
                <Year>2011</Year>
                <Period>1</Period>
                <FormData>
                    <TextField id="01">01/01/2011</TextField>
                    <TextField id="02">31/12/2011</TextField>
                    <TextField id="03">EUR</TextField>
                    <NumericField id="109">12805,43</NumericField>
                    <NumericField id="110">10986,88</NumericField>
                    ...
                    ...
                    <NumericField id="405">60914,26</NumericField>
                    <NumericField id="406">87104,32</NumericField>
                </FormData>
            </Declaration>
            <Declaration type="CA_COMPP" model="1" language="EN">
                <Year>2011</Year>
                <Period>1</Period>
                <FormData>
                    <TextField id="01">01/01/2011</TextField>
                    <TextField id="02">31/12/2011</TextField>
                    <TextField id="03">EUR</TextField>
                    <NumericField id="639">0,00</NumericField>
                    <NumericField id="640">0,00</NumericField>
                    ...
                    ...
                    <NumericField id="737">27707,11</NumericField>
                    <NumericField id="738">52429,42</NumericField>
                </FormData>
            </Declaration>
        </Declarer>
    </Declarations>
</eCDFDeclarations>
```

The element of the last declarer is followed by closing tags for the elements « Declarations » and « eCDFDeclarations ».

5. Content of the file

The eCDF XML schema in its latest version is available for download as an XSD file in the menu «Developer / Documentation».

The following aspects related to content are not specified in the XML schema and cannot be validated against it.

- Only declarations corresponding to the forms, models and reference years listed in the menu "Declarer / Documentation" can be included in an eCDF-XML file of version 2.0.
- It is recommended to insert the elements corresponding to fields under « FormData » in the same order as they appear on the form. This practice makes it easier to read the XML file when an error is reported. If the form contains tables, they must however always appear after the individual fields in the XML file.
- In a « DeclarationGroup », the « MappingTable » must always be in the last position.
- A field can appear in a declaration only once. It is therefore impossible for two «TextField», «NumericField» or «Choice» elements within the same « FormData » element to have the same value in their « id » attribute, unless in a table.
- The value format within the « TextField » fields is not controlled by the XML schema but only in a second step by the eCDF validation engine. Potential problems in the format of the file generated by your interface are therefore not detectable by a validity check against the eCDF schema. This is particularly true for dates.
- For annual accounts, the beginning of the year must always be later than or equal to 01/01/2011. Previous years cannot be deposited into the eCDF platform.
- For annual accounts, the eCDF platform does not accept shortened financial years for 2011, that is to say financial years with an end date earlier than 31/12/2011.
- For VAT declarations, the declaration year has to be 2015 or later.
- For VAT recapitulative statements, the declaration year has to be 2017 or later.

The eCDF area dedicated to interface developers provides a function to contact the eCDF team of the CTIE and send them a file to be tested. This may be used if the file was rejected by the transfer test function and the documentation available on the site did not help to find the reason of the problem. More information on this subject is available in the « User guide for the interface developer's area ».

6. Practical tips for programming the XML interface

The eCDF support team is regularly contacted by XML interface users whose XML files are rejected by eCDF, due to problems that could easily be avoided by a few simple improvements in the XML interfaces and especially their configuration pages.

6.1. Use the same terms that are also used in eCDF

By using the same terms in the configuration page of your XML interface as those used in eCDF and especially in error messages, this will allow the user to understand which parameter needs to be corrected.

eCDF term	Explanation and synonyms
Agent	This is the person or company using your accounting software. alias : fiduciary; ….
Declarer	This is the person on whose behalf a return is created. alias : client ; (previously: mandant)

6.2. Group and lock agent settings

The parameters to configure for the agent are as follows:

- prefix eCDF
- personnel id number (matricule)
- RCSL number (not available in all cases)
- VAT number (not available in all cases)

These settings identify the agent and should be configured once when installing the XML interfaces. It is therefore advisable to

- to save these parameters in order to avoid a new entry each time you use them
- to lock these parameters in order to prevent a change by mistake
- to indicate on the screen that these parameters are those of the agent who uses the accounting tool and must not be changed

6.3. Facilitate the configuration of the declarant

The parameters to be configured for the declarant are as follows:

- personnel id number (matricule)
- RCSL number (not available in all cases)
- VAT number (not available in all cases)

To avoid unnecessary errors in the XML file, it is recommended at least

- to provide a checkbox that the user can click to indicate if an RCSL or VAT number does not exist. Asking the user to manually enter "NE" in the RCSL or VAT field may result in input errors with values different from "NE" which are not allowed.
- to control that each parameter is indicated, either as an identification number or as a "NE". This prevents the mandatory information from being missing which results in the file being rejected.

6.4. Create a new file reference each time a file is created

Each file reference can only be transferred into eCDF only once. Even if a first incorrect XML upload is deleted by the agent, this same file reference cannot be used again.

It is therefore recommended that the XML interface automatically creates a new file reference and name for each file created.