

Data Export User Manual

**Data transferred to the auditor program in
XML format**

Table of contents

1	Content requirements of the XML	3
1.1	Overview	3
1.2	Defined Types	3
1.3	Restrictions	3
1.4	Defined Structure	5
1.4.1	XML Data	6
1.4.2	Firm Information	7
1.4.3	Parameters	9
1.4.4	Verification Numbers	11
1.4.5	Journals	12
1.4.6	Periods	13
1.4.7	Account Numbers	14
1.4.8	Partners	16
1.4.9	Persons Recording the Entries	17
1.4.10	General Ledger Source Documents	18
1.4.11	General Ledger Entries	20
2	XML Template of the Defined Structure (XSD)	23
3	Examples for Preparing the XML File	34
3.1	Summarizing Example	34
3.2	Example of a Ledger Entry with VAT	38
3.3	Example of a Cancellation Ledger Entry Pair	40
4	Support for Creating the XML File	42
4.1	Technical Suggestions	42
4.2	Personal Support	42
5	Tracking Changes Compared to the Previous Version	43
5.1	Documentation	43
5.2	XML Structure	43
6	About XML	44
6.1	Special Characters	44

1 Content requirements of the XML

1.1 Overview

The first line of an XML is always `<?xml version="1.0" encoding="..."?>`, where `version` describes the version of the XML structure to the XML interpreter and `encoding` describes the character set of the text content of the XML file. The encoding can be UTF-8, ISO-8859-2, ISO-8859-1, etc., depending on which was used when the file was created. If the accentuated characters cannot be displayed properly, the character set must be changed.

The structure of the XML is made up of tags and tag pairs. If the tag includes a value, then the value is between the opening tag and the closing tag:

`<Nev>Something</Nev>`, while if there is no value, then a single tag is both the opening and the closing ones: `<Nev/>`. This is the same structure as `<Nev></Nev>`, but the former takes less space. The tags are case-sensitive, which means that the defined tags are strictly set within the document. The following format is incorrect: `<Nev>Something</nev>`.

1.2 Defined Types

You can define values with the following types:

- `Char` type can contain alphanumeric characters.
- `Integer` type can contain numeric characters only.
- `Numeric` type can contain a numeric value with either a comma (`,`), or a colon (`. `) marking the decimal point. Do not mark the thousands separator.
- `Logical` type is a logical value, either Y or N.
- `Date` type uses the following format: 2010-02-19.
- `DateTime` type uses an exact time in the following format: 2010-02-19 14:58:53.

1.3 Restrictions

For data processing the tags must be broken into separate lines.

The tag must be given even if it contains no value, otherwise the structure is incomplete and the document does not comply with the preset rules.

A value must be provided for the tags that are marked with a “Yes” in the “Mandatory” column of the structure description. In case the relevant tag does not have a value then use the following:

Char	“no data” or “n.a.”
Integer Numeric	“0”
Logical	“N”
Date	“1900-01-01”
DateTime	“1900-01-01 00:00:00”

Developers are encouraged to use the “Audit XML export” as the output file name for consistent reference, if possible.

1.4 Defined Structure

We expect the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<Adatok xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<XMLAdatok>...</XMLAdatok>
<Cegadatok>...</ Cegadatok >
<Parameterek>...</Parameterek>
<Ellenorzes>...</Ellenorzes>
<Naplok>...</Naplok >
<Idoszakok>...</Idoszakok>
<Szamlaszamok>...</Szamlaszamok>
<Partnerek>...</Partnerek>
<Rogzitok>...</Rogzitok>
<FkBizonylatok>...</FkBizonylatok>
<FkTetelek>...</FkTetelek>
</Adatok>
```

Find the description of the tag pairs in the following as well as a short example in chapter 3.

Tag	Mandatory	Type	Length	Description
XMLAdatok	yes			Details related to the created file.
Cegadatok	yes			Details related to the booked firm (hereinafter referred to as firm)
Parameterek				Unique properties related to the bookkeeping of the firm.
Ellenorzes	yes			Count of data in a given segment.
Naplok				Journals of the firm's general ledger.
Idoszakok				Accounting periods of the firm.
Szamlaszamok	yes			General ledger accounts of the firm.
Partnerek				Partners (buyers, suppliers) of the firm.
Rogzitok	yes			Persons entering the data for bookkeeping.
FkBizonylatok	yes			Source documents of the

				general ledger.
FkTetelek	yes			Items of the source documents of the general ledger.

1.4.1 XML Data

The XMLAdatok tag contains the information related to creating and using the file.

Example:

```
<XMLAdatok>
  <Verzio>1.0.1.0</Verzio>
  <LetrehozoProgram>
    <Nev>DimSQL</Nev>
    <Verzio>3.42</Verzio>
  </LetrehozoProgram>
  <Letrehozva>2010-04-16 14:51:42</Letrehozva>
</XMLAdatok>
```

Description:

Tag	Mandatory	Type	Length	Description
Verzio	yes	Char	20	Version of the XML structure (1.0.1.0 in this example).
LetrehozoProgram	yes			Information about the program that created the XML file (more details can be found below).
Letrehozva	yes	DateTime		The time when the XML file was created.

1.4.1.1 Creator Program

Description:

Tag	Mandatory	Type	Length	Description
Nev	yes	Char	50	Name of the program that created the XML file.
Verzio		Char	20	Version of the program that created the XML file

1.4.2 Firm Information

Information about the firm is important to identify the data in the XML file and to make use of the ledger items so most tags are mandatory.

Example:

```
<Cegadatok>
  <Nev>Bemutato Kft.</Nev>
  <Adoszam>12345678-1-23</Adoszam>
  <KezdoDatum>2010-01-01</KezdoDatum>
  <VegsoDatum>2010-12-31</VegsoDatum>
  <Penznem>HUF</Penznem>
  <PenzEgyseg>MNB alapegység</PenzEgyseg>
  <Cim>
    <Orszag>Magyarország</Orszag>
    <Telepules>Budapest</Telepules>
    <Irszam>1117</Irszam>
    <KozterNev>Nádorliget</KozterNev>
    <KozterJell>utca</KozterJell>
    <Hazzsam>8</Hazzsam>
    <Epulet>a</Epulet>
    <Lepcsohaz />
    <Emelet />
    <Ajto />
  </Cim>
  <KapcsolatTarto>Gipsz Jakab</KapcsolatTarto>
  <Telefonszam>+36-1/123-45-67</Telefonszam>
</Cegadatok>
```

Description of contents:

Tag	Mandatory	Type	Length	Description
Nev	yes	Char	100	Name of the firm.
Adoszam	yes	Char	13	Domestic tax number of the firm.
KezdoDatum	yes	Date	10	Starting date of the firm's business year.
VegsoDatum	yes	Date	10	End date of the firm's business year.
Penznem	yes	Char	5	Default currency.
PenzEgyseg	yes	Char	15	Unit of the default values (MNB default unit, thousand, million).
Cim				Address of the firm.
KapcsolatTarto		Char	50	Contact person in the firm regarding the audit.
Telefonszam		Char	15	Phone number of the contact person.

1.4.2.1 Address

Description of contents:

Tag	Mandatory	Type	Length	Description
Orszag	yes	Char	40	Country.
Telepules	yes	Char	40	Town/city.
Irszam	yes	Char	10	Post code.
KozterNev	yes	Char	40	Name of the public place in the address.
KozterJell	yes	Char	30	Type of the public place in the address. (e.g.: street, road, place, etc.)
Hazszam	yes	Char	7	House number.
Epulet		Char	3	Name of the building.
Lepcsohaz		Char	3	Number of the stairwell.
Emelet		Char	4	Floor number.
Ajto		Char	4	Door number.

1.4.3 Parameters

The account numbers of the parameters contain data that help to determine the type of the ledger account number. It is not mandatory to fill out these account numbers, but it makes the work of the auditor easier. It is recommended to fill them out if the number is available in the program.

Example:

```
<Parametererek>
  <SzamlaSzamok>
    <Vevo>311*,312*,</Vevo>
    <Szallito>454*,455*</Szallito>
    <Penztar>381*,382*</Penztar>
    <Bank>384*,386*</Bank>
    <Deviza>312*,455*,382*,386*</Deviza>
    <BeAFA>467</BeAFA>
    <ViAFA>466</ViAFA>
    <KoAFA />
  </SzamlaSzamok>
</Parametererek>
```

Description of contents:

Tag	Mandatory	Type	Length	Description
SzamlaSzamok				Special account numbers (further details below).

1.4.3.1 Account Numbers

A group of ledger account numbers can be defined to be dealt with in a special way by the program or that need special attention during the auditing process. Such groups are the accounts receivable, accounts payable, cash, bank and VAT accounts.

One or more account numbers can be defined as the value of the tag. In case more account numbers belong to the tag, they must be separated with a ‘,’ (comma), without any spaces in between (e.g.: in case of banks: 3841,3842,3861,3862). If more than one account number start with the same numbers, then the list can be defined with an ‘*’ (asterix) character (e.g.: in case of banks: 384*,386*). In this case, every account number that begins with the given numbers are included in the list.

Description of contents:

Tag	Mandatory	Type	Length	Description
Vevo		Char	255	Accounts receivable
Szallito		Char	255	Accounts payable.
Penztar		Char	255	Petty cash accounts.

Bank		Char	255	Bank accounts.
Deviza		Char	255	Foreign exchange accounts.
BeAFA		Char	255	VAT payable accounts.
ViAFA		Char	255	VAT recoverable accounts.
KoAFA		Char	255	Accounts of compensatory premium.

1.4.4 Verification Numbers

Verification numbers are mandatory as they are necessary for the extraction of the data. The value of these tags gives the cardinality of the sub tags (the number of records in a given table) within other XML tags.

Example:

```
<Ellenorzes>
  <Naplok>12</Naplok>
  <Idoszakok>14</Idoszakok>
  <Szamlaszamok>9259</Szamlaszamok>
  <Partnerek>8338</Partnerek>
  <Rogzitok>9</Rogzitok>
  <FkBizonylatok>49496</FkBizonylatok>
  <FkTetelek>133946</FkTetelek>
</Ellenorzes>
```

Description of contents:

Tag	Mandatory	Type	Length	Description
Naplok	yes	Integer		Number of journals.
Idoszakok	yes	Integer		Number of periods.
Szamlaszamok	yes	Integer		Number of account numbers.
Partnerek	yes	Integer		Number of partners.
Rogzitok	yes	Integer		Number of persons recording the entries.
FkBizonylatok	yes	Integer		Number of general ledger source documents.
FkTetelek	yes	Integer		Number of general ledger entries.

1.4.5 Journals

The “Journal” defines in which journal the document was recorded in the general ledger and which module it came to the ledger from (e.g.: bank journal, adjustments journal, journal of invoices).

This value is not mandatory as not all programs contain it. In case the Naplok identification was filled out at the source documents, then the ID and the title must be given here, too.

The technical reason to separate the journals from the source documents is to speed up the identification of the journals, thus reduce the size of the file.

Example:

```
<Naplok>
  <Naplo>
    <Kod>1</Kod>
    <Nev>Nyitó</Nev>
  </Naplo>
  <Naplo>
    <Kod>2</Kod>
    <Nev>Bank</Nev>
  </Naplo>
  <Naplo>
    <Kod>3</Kod>
    <Nev>Pénztár</Nev>
  </Naplo>
  <Naplo>
    <Kod>4</Kod>
    <Nev>Vevő</Nev>
  </Naplo>
  <Naplo>
    <Kod>5</Kod>
    <Nev>Szállító</Nev>
  </Naplo>
  <Naplo>
    <Kod>13</Kod>
    <Nev>Záró</Nev>
  </Naplo>
</Naplok>
```

Description of contents:

Tag	Mandatory	Type	Length	Description
Kod	yes	Integer (PK)		Identifier of the journal; the value of the FkBizonylatok / Biz / Naplo tag.
Nev	yes	Char	20	Name of the log.

1.4.6 Periods

Period of time (typically a month) in a given business year that binds the source documents into an accounting period.

This value is not mandatory as not all programs contain it. In case the `Idoszakok` identification was filled out at the source documents, then the ID and the title must be given here, too.

The technical reason to separate the periods from the source documents is to speed up the identification of the logs, thus reduce the size of the file.

Example:

```
<Idoszakok>
  <Idoszak>
    <Kod>0</Kod>
    <Nev>NYITÁS</Nev>
  </Idoszak>
  <Idoszak>
    <Kod>1</Kod>
    <Nev>január</Nev>
  </Idoszak>
  <Idoszak>
    <Kod>2</Kod>
    <Nev>február</Nev>
  </Idoszak>
  <Idoszak>
    <Kod>12</Kod>
    <Nev>december</Nev>
  </Idoszak>
  <Idoszak>
    <Kod>13</Kod>
    <Nev>ZÁRÁS</Nev>
  </Idoszak>
</Idoszakok>
```

Description of contents:

Tag	Mandatory	Type	Length	Description
Kod	yes	Integer (PK)		Identifier of the period; the value of the <code>FkBizonylatok / Biz / Idoszak</code> tag.
Nev	yes	Char		Name of the period.

1.4.7 Account Numbers

The account number identifies the debit and credit accounts of the entries.

This value is mandatory at the entries (Tartozik, Kovetel). The Tartozik (Debit) and Kovetel (Credit) identification must be defined here as well including the name of the account, but it is not mandatory to add the alien (foreign) code and alien (foreign) title.

The Szamlaszam (Account number) tag defines the individual account numbers and the Szamlaszamok (Account numbers) tag binds the account numbers together.

The technical reason to separate the account numbers from the entries is to speed up the identification of the logs, thus reduce the size of the file.

Example:

```
<Szamlaszamok>
  <Szamlaszam>
    <Kod>111</Kod>
    <TKod>0111</TKod>
    <Nev>0-ra leirt immateriális javak</Nev>
    <ITKod />
    <INev />
  </Szamlaszam>
  <Szamlaszam>
    <Kod>311</Kod>
    <TKod />
    <Nev>Vevők (belföldi)</Nev>
    <ITKod />
    <INev />
  </Szamlaszam>
  <Szamlaszam>
    <Kod>4791</Kod>
    <TKod />
    <Nev>Osztalék</Nev>
    <ITKod />
    <INev>DIVIDENDE</INev>
  </Szamlaszam>
</Szamlaszamok>
```

Description of contents:

Tag	Mandatory	Type	Length	Description
Kod	yes	Integer (PK)		Identifier of the account; the value of the FkTetelek / Tet / Tartozik or Kovetel tag.
TKod		Char	10	Text identification of the account number (for

				systems where the ID is not integer type).
Nev	yes	Char	50	Name of the account.
ITKod		Char	10	Text identification of the account. in a foreign language
INev		Char	50	Name of the account in a foreign language.

1.4.8 Partners

The partner code identifies the partner in case of entries that concern a customer or a supplier. In such cases it is mandatory to fill out the `Partner` value.

If the `Partner` field was filled out when recording the entry, then the identification and the name must be given here as well.

The technical reason to separate the partners from the entries is to speed up the identification of the partners, thus reduce the size of the file.

Example:

```
<Partnerek>
  <Partner>
    <Kod>1001</Kod>
    <TKod />
    <Nev>Pirker Kft.</Nev>
  </Partner>
  <Partner>
    <Kod>1003</Kod>
    <TKod />
    <Nev>Gránit 2000 Kft.</Nev>
  </Partner>
  <Partner>
    <Kod>5001</Kod>
    <TKod>P- 5001</TKod>
    <Nev>Gallup GmbH</Nev>
  </Partner>
  <Partner>
    <Kod>5002</Kod>
    <TKod />
    <Nev>Mixer Ltd</Nev>
  </Partner>
</Partnerek>
```

Description of contents:

Tag	Mandatory	Type	Length	Description
Kod	yes	Integer (PK)		Identifier of the partner; the value of the <code>FkTetelek / Tet / Partner</code> tag.
TKod		Char	10	Text identification of the account (for systems where the ID is not integer type).
Nev	yes	Char	255	Name of the partner.

1.4.9 Persons Recording the Entries

The code of the person recording the entries identifies the person itself, and possibly the system as well.

It is mandatory to fill out this field (Rogzito). The same Rogzito ID must be given here as the one given at the recorded entry.

Details of the individuals are included under the Rogzito tag, while Rogzitok contains all the persons recording entries.

The technical reason to separate these details from the entries is to speed up the identification of the persons recording the entries, thus reduce the size of the file.

Example:

```
<Rogzitok>
  <Rogzito>
    <Kod>1</Kod>
    <TKod />
    <Nev>Nincs Adat</Nev>
  </Rogzito>
  <Rogzito>
    <Kod>2</Kod>
    <TKod>GJAKAB</TKod>
    <Nev>Gipsz Jakab</Nev>
  </Rogzito>
</Rogzitok>
```

Description of contents:

Tag	Mandatory	Type	Length	Description
Kod	yes	Integer (PK)		Identifier of the person recording the entry; the value of the FkTetelek / Tet / Rogzito tag.
TKod		Char	10	Text identification of the person recording the entry (for systems where the ID is not integer type).
Nev	yes	Char	50	Name of the person recording the entry.

1.4.10 General Ledger Source Documents

Contains the source documents of the general ledger with the identifications described earlier.

[Source Documents](#) are included in the `FkBizonnylatok` tag, while the details of each source document are defined under the `Biz` tag.

Example:

```
<FkBizonnylatok>
  <Biz>
    <BizID>-999999</BizID>
    <Naplo>1</Naplo>
    <BizSzam>NYITÁS</BizSzam>
    <Datum>2010-01-01</Datum>
    <Idoszak>0</Idoszak>
  </Biz>
  <Biz>
    <BizID>1</BizID>
    <Naplo>10</Naplo>
    <BizSzam>2010/001</BizSzam>
    <Datum>2010-01-15</Datum>
    <Idoszak>1</Idoszak>
  </Biz>
</FkBizonnylatok>
```

Description of contents:

Tag	Mandatory	Type	Length	Description
BizID	yes	Integer (PK)		Identification of the document containing the entries (program-generated sequential number, unique to the document).
Naplo		Integer		Identifier of the journal defining in which journal the document was recorded in the general ledger and which submodule it came from to the general ledger.
BizSzam	yes	Char	50	Number of the document (e.g.: number of the account receivable, registration number of the account payable).
Datum	yes	Date		Date of the transaction. The date of issuing the invoice

				in case of accounts payable or receivable.
Idoszak		Integer		The identification of the period within an accounting year that categorizes the booked document into a given period.

1.4.11 General Ledger Entries

Contains the items of the general ledger with the previously described identifications.

Data for the individual entries are within the `Tet` tag, while the entries are included in the `FkTetelek` tag.

Example:

```
<FkTetelek>
  <Tet>
    <BizID>-1</BizID>
    <TetID>1730</TetID>
    <Szoveg>ÉVES NYITÁS</Szoveg>
    <Tartozik>312001</Tartozik>
    <Kovetel>491</Kovetel>
    <Osszeg>32500</Osszeg>
    <DevOsszeg>125</DevOsszeg>
    <DevNem>EUR</DevNem>
    <AfaAlap>130000</AfaAlap>
    <AfaKulcs>25</AfaKulcs>
    <Partner>1001</Partner>
    <PuAzo>2009/00006</PuAzo>
    <TeljDatum>2009-08-21</TeljDatum>
    <FizHatarido>2009-08-21</FizHatarido>
    <Szt />
    <SztTetID />
    <Rogzito>1</Rogzito>
    <Rogzitve>2006-05-28 13:54:56</Rogzitve>
  </Tet>
</FkTetelek>
```

Description of contents:

Tag	Mandatory	Type	Length	Description
BizID	Yes	Integer (FK)		Identification of the document (a sequential number generated by the software, unique to the document).
TetID	Yes	Integer (PK)		Identification of the accounting entry (a sequential number generated by the software, unique to the entry).
Szoveg	Yes	Char	50	Text comment to the accounting entry.
Tartozik	Yes	Integer (FK)		General Ledger account number of the debit side of the entry.
Kovetel	Yes	Integer		General Ledger account

		(FK)		number of the credit side of the entry.
Osszeg	Yes	Numeric		Amount of the entry. If there is VAT, the first item is the net amount; the second item is the VAT.
DevOsszeg	*	Numeric		In case of foreign currency the HUF amount translated into the foreign currency.
DevNem	*	Char	5	In case of foreign currency denominated transaction the type of the foreign currency.
AfaAlap	*	Numeric		In case of VAT-content entry the net amount which forms base for the VAT calculation.
AfaKulcs	*	Char	10	Alphanumeric (that is, it may include numbers and letters of the alphabet as well) code of VAT rate (e.g.: 0, 5, 15, 18, 20, 25, M, EM, AHK).
Partner	*	Integer (FK)		Identification of the partner in case of accounts receivable or accounts payable.
PuAzo	*	Char	20	Code for matching the invoices with the related payments at the accounts payable or accounts receivable (e.g. invoice number), which is base for matching the recording and payment posting.
TeljDatum	*	Date		At the accounts receivables or accounts payables the fulfillment date on the invoice (according to the Hungarian VAT-law) used for the purposes of the VAT-return. To be used when recording or modifying the receivable or payable, not necessarily when recording actual payment.
FizHatarido	*	Date		Payment deadline in case of accounts receivable or

				accounts payable. To be used when recording or modifying the receivable or payable, not necessarily when recording actual payment.
Szt	*	Logical	1	In case of entry cancellation the value of the tag is 'I', in every other case it is 'N' or empty meaning the entry is valid and uncanceled.
SztTetID	*	Integer		The TetID of the cancelled entry, which becomes invalid by this entry.
Rogzito	Yes	Integer (FK)		Identification of the person recording the entry.
Rogzitve	Yes	DateTime		Date and time of recording the entry.

*: only to be filled in if such data exists as a result of the economic transaction.

2 XML Template of the Defined Structure (XSD)

The following template can be used to prepare and check the structure of the XML file, without checking the contents.

That is, the template does not verify whether the value of an Integer (PK) type is indeed unique in the given segment, or that the value of an Integer (FK) type are defined in the referred segment, etc.

The template can be downloaded from the MKVK website, where this document is also located.

```
<?xml version="1.0" encoding="ISO-8859-2"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:simpleType name="NemKotelezoIntegerTipus">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="|([0-9]+)|[-]([0-9]+)"/>
      <xsd:maxLength value="10"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="NemKotelezoDecimalTipus">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="|([0-9]+)|[-]([0-9]+)|([0-9]+)*[.,]([0-9]+)|[-]([0-9]+)*[.,]([0-9]+)"/>
      <xsd:maxLength value="20"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="NemKotelezoDateTipus">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="|((19|20)[0-9]{2})-((0[1-9])|(1[0-2]))-((0[1-9])|([1,2][0-9])|(3[0,1]))"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="KotelezoStringTipus">
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="KotelezoDecimalTipus">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="([0-9]+)|[-]([0-9]+)|([0-9]+)*[.,]([0-9]+)|[-]([0-9]+)*[.,]([0-9]+)"/>
      <xsd:maxLength value="20"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="KotelezoDateTipus">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="((19|20)[0-9]{2})-((0[1-9])|(1[0-2]))-((0[1-9])|([1,2][0-9])|(3[0,1]))"/>
    </xsd:restriction>
  </xsd:simpleType>

</xsd:schema>
```

```

<xsd:simpleType name="KotelezoDateTimeTipus">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="((19|20)[0-9]{2})-((0[1-9])|(1[0-2]))-((0[1-9])|([1,2][0-9])|(3[0,1]))( |T)(([0,1][0-9])|(2[0-3])):([0-5][0-9]):([0-5][0-9])"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name="Adatok">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="XMLAdatok" type="XMLAdatokTipus" />
      <xsd:element name="Cegadatok" type="CegadatokTipus" />
      <xsd:element name="Parameterek" type="ParameterekTipus" />
      <xsd:element name="Ellenorzes" type="EllenorzesTipus" />
      <xsd:element name="Naplok" type="NaplokTipus" />
      <xsd:element name="Idoszakok" type="IdoszakokTipus" />
      <xsd:element name="Szamlaszamok" type="SzamlaszamokTipus" />
      <xsd:element name="Partnerek" type="PartnerekTipus" />
      <xsd:element name="Rogzitok" type="RogzitokTipus" />
      <xsd:element name="FkBizonylatok" type="FkBizonylatokTipus" />
      <xsd:element name="FkTetelek" type="FkTetelekTipus" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="XMLAdatokTipus">
  <xsd:sequence>
    <xsd:element name="Verzio" fixed="1.0.1.0">
      <xsd:simpleType>
        <xsd:restriction base="KotelezoStringTipus">
          <xsd:maxLength value="20"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="LetrehozoProgram"
type="LetrehozoProgramTipus"/>
    <xsd:element name="Letrehozva" type="KotelezoDateTimeTipus"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="LetrehozoProgramTipus">
  <xsd:sequence>
    <xsd:element name="Nev">
      <xsd:simpleType>
        <xsd:restriction base="KotelezoStringTipus">
          <xsd:maxLength value="50"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Verzio">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:maxLength value="20"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```



```

        </xsd:simpleType>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CegadatokTipus">
    <xsd:sequence>
        <xsd:element name="Nev">
            <xsd:simpleType>
                <xsd:restriction base="KotelezoStringTipus">
                    <xsd:maxLength value="100"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="Adoszam">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:pattern value="[0-9]{8}-[0-9]{1}-[0-9]{2}"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="KezdoDatum" type="KotelezoDateTipus" />
        <xsd:element name="VegsoDatum" type="KotelezoDateTipus" />
        <xsd:element name="Penznem">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="HUF"/>
                    <xsd:enumeration value="EUR"/>
                    <xsd:enumeration value="USD"/>
                    <xsd:enumeration value="GBP"/>
                    <xsd:enumeration value="JPY"/>
                    <xsd:enumeration value="CHF"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="PenzEgyseg">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="száz"/>
                    <xsd:enumeration value="ezer"/>
                    <xsd:enumeration value="millió"/>
                    <xsd:enumeration value="MNB alapegység"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="Cim" type="CimTipus" />
        <xsd:element name="KapcsolatTarto">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="50"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="Telefonszam">

```

```

    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:maxLength value="15"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CimTipus">
  <xsd:sequence>
    <xsd:element name="Orszag">
      <xsd:simpleType>
        <xsd:restriction base="KotelezoStringTipus">
          <xsd:maxLength value="40"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Telepules">
      <xsd:simpleType>
        <xsd:restriction base="KotelezoStringTipus">
          <xsd:maxLength value="40"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Irszam">
      <xsd:simpleType>
        <xsd:restriction base="KotelezoStringTipus">
          <xsd:maxLength value="10"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="KozterNev">
      <xsd:simpleType>
        <xsd:restriction base="KotelezoStringTipus">
          <xsd:maxLength value="40"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="KozterJell">
      <xsd:simpleType>
        <xsd:restriction base="KotelezoStringTipus">
          <xsd:maxLength value="30"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Hazszam">
      <xsd:simpleType>
        <xsd:restriction base="KotelezoStringTipus">
          <xsd:maxLength value="7"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

```

<xsd:element name="Epulet">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="3"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="Lepcsohaz">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="4"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="Emelet">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="4"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="Ajto">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="|([0-9]+)"/>
      <xsd:maxLength value="4"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ParameterekTipus">
  <xsd:sequence>
    <xsd:element name="SzamlaSzamok" type="ParSzamlaSzamokTipus" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ParSzamlaSzamokTipus">
  <xsd:sequence>
    <xsd:element name="Vevo">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:maxLength value="255"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Szallito">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:maxLength value="255"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

```

</xsd:element>
<xsd:element name="Penztar">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="255"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="Bank">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="255"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="Deviza">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="255"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="BeAFA">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="255"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="ViAFA">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="255"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="KoAFA">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="255"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="EllenorzesTipus">
  <xsd:sequence>
    <xsd:element name="Naplok" type="xsd:nonNegativeInteger" />
    <xsd:element name="Idoszakok" type="xsd:nonNegativeInteger" />
    <xsd:element name="Szamlaszamok" type="xsd:positiveInteger" />
    <xsd:element name="Partnerek" type="xsd:nonNegativeInteger" />
    <xsd:element name="Rogzitok" type="xsd:positiveInteger" />
  </xsd:sequence>
</xsd:complexType>

```

```

        <xsd:element name="FkBizonylatok" type="xsd:positiveInteger" />
        <xsd:element name="FkTetelek" type="xsd:positiveInteger" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="NaplokTipus">
    <xsd:sequence>
        <xsd:element name="Naplo" type="NaploTipus" minOccurs="0"
maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="NaploTipus">
    <xsd:sequence>
        <xsd:element name="Kod" type="xsd:integer" />
        <xsd:element name="Nev">
            <xsd:simpleType>
                <xsd:restriction base="KotelezoStringTipus">
                    <xsd:maxLength value="20"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="IdoszakokTipus">
    <xsd:sequence>
        <xsd:element name="Idoszak" type="IdoszakTipus" minOccurs="0"
maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="IdoszakTipus">
    <xsd:sequence>
        <xsd:element name="Kod" type="xsd:integer" />
        <xsd:element name="Nev">
            <xsd:simpleType>
                <xsd:restriction base="KotelezoStringTipus">
                    <xsd:maxLength value="20"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SzamlaszamokTipus">
    <xsd:sequence>
        <xsd:element name="Szamlaszam" type="SzamlaszamTipus"
minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SzamlaszamTipus">

```

```

<xsd:sequence>
  <xsd:element name="Kod" type="xsd:integer" />
  <xsd:element name="TKod">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:maxLength value="10"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="Nev">
    <xsd:simpleType>
      <xsd:restriction base="KotelezoStringTipus">
        <xsd:maxLength value="50"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="ITKod">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:maxLength value="10"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="INev">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:maxLength value="50"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="PartnerekTipus">
  <xsd:sequence>
    <xsd:element name="Partner" type="PartnerTipus" minOccurs="0"
maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="PartnerTipus">
  <xsd:sequence>
    <xsd:element name="Kod" type="xsd:integer" />
    <xsd:element name="TKod">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:maxLength value="10"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Nev">
      <xsd:simpleType>
        <xsd:restriction base="KotelezoStringTipus">

```

```

        <xsd:maxLength value="255"/>
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="RogzitokTipus">
    <xsd:sequence>
        <xsd:element name="Rogzito" type="RogzitoTipus" minOccurs="1"
maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="RogzitoTipus">
    <xsd:sequence>
        <xsd:element name="Kod" type="xsd:integer" />
        <xsd:element name="TKod">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="10"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="Nev">
            <xsd:simpleType>
                <xsd:restriction base="KotelezoStringTipus">
                    <xsd:maxLength value="50"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FkBizonylatokTipus">
    <xsd:sequence>
        <xsd:element name="Biz" type="FkBizTipus" minOccurs="1"
maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FkBizTipus">
    <xsd:sequence>
        <xsd:element name="BizID" type="xsd:integer" />
        <xsd:element name="Naplo" type="NemKotelezoIntegerTipus" />
        <xsd:element name="BizSzam">
            <xsd:simpleType>
                <xsd:restriction base="KotelezoStringTipus">
                    <xsd:maxLength value="50"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="Datum" type="KotelezoDateTipus" />
    </xsd:sequence>
</xsd:complexType>

```

```

    <xsd:element name="Idoszak" type="NemKotelezoIntegerTipus" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FkTetelekTipus">
  <xsd:sequence>
    <xsd:element name="Tet" type="FkTetTipus" minOccurs="1"
maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FkTetTipus">
  <xsd:sequence>
    <xsd:element name="BizID" type="xsd:integer" />
    <xsd:element name="TetID" type="xsd:integer" />
    <xsd:element name="Szoveg">
      <xsd:simpleType>
        <xsd:restriction base="KotelezoStringTipus">
          <xsd:maxLength value="50"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Tartozik" type="xsd:integer" />
    <xsd:element name="Kovetel" type="xsd:integer" />
    <xsd:element name="Osszeg" type="KotelezoDecimalTipus" />
    <xsd:element name="DevOsszeg" type="NemKotelezoDecimalTipus" />
    <xsd:element name="DevNem">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:maxLength value="5"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="AfaAlap" type="NemKotelezoDecimalTipus" />
    <xsd:element name="AfaKulcs">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:maxLength value="10"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Partner" type="NemKotelezoIntegerTipus" />
    <xsd:element name="PuAzo">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:maxLength value="20"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="TeljDatum" type="NemKotelezoDateTipus" />
    <xsd:element name="FizHatarido" type="NemKotelezoDateTipus" />
    <xsd:element name="Szt">
      <xsd:simpleType>

```



```
<xsd:restriction base="xsd:string">
  <xsd:maxLength value="1"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="SztTetID" type="NemKotelezoIntegerTipus" />
<xsd:element name="Rogzito" type="xsd:integer" />
<xsd:element name="Rogzitve" type="KotelezoDateTimeTipus" />
</xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

3 Examples for Preparing the XML File

3.1 Summarizing Example

Please, do not consider the accounting in the example! This example contains only two general ledger entries with the relevant data and as such, the accounting contents are incomplete. The reason of the example is to show the structure of the XML!

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<Adatok xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <XMLAdatok>
    <Verzio>1.0.1.0</Verzio>
    <LetrehozoProgram>
      <Nev>DimSQL</Nev>
      <Verzio>3.40</Verzio>
    </LetrehozoProgram>
    <Letrehozva>2010-02-15 11:43:52</Letrehozva>
  </XMLAdatok>
  <Cegadatok>
    <Nev>Default Llc.</Nev>
    <Adoszam>12345678-1-23</Adoszam>
    <KezdoDatum>2010-01-01</KezdoDatum>
    <VegsoDatum>2010-12-31</VegsoDatum>
    <Penznem>HUF</Penznem>
    <PenzEgyseg>MNB default unit</PenzEgyseg>
    <Cim>
      <Orszag>Hungary</Orszag>
      <Telepules>Budapest</Telepules>
      <Irszam>1117</Irszam>
      <KozterNev>Nádorliget</KozterNev>
      <KozterJell>street</KozterJell>
      <Hazzsam>8</Hazzsam>
      <Epulet>a</Epulet>
      <Lepcsohaz/>
      <Emelet/>
      <Ajto/>
    </Cim>
    <KapcsolatTarto>John Doe</KapcsolatTarto>
    <Telefonszam>+36-1/123-45-67</Telefonszam>
  </Cegadatok>
  <Parameterek>
    <SzamlaSzamok>
      <Vevo>311,312</Vevo>
      <Szallito>454,455</Szallito>
      <Penztar>381,382</Penztar>
      <Bank>384,386</Bank>
      <Deviza>312,455,382,386</Deviza>
      <BeAFA>467</BeAFA>
      <ViAFA>466</ViAFA>
      <KoAFA></KoAFA>
    </SzamlaSzamok>
  </Parameterek>
</Adatok>
```

```
</SzamlaSzamok>
</Parameterek>
<Ellenorzes>
  <Naplok>2</Naplok>
  <Idoszakok>1</Idoszakok>
  <Szamlaszamok>4</Szamlaszamok>
  <Partnerek>1</Partnerek>
  <Rogzitok>2</Rogzitok>
  <FkBizonylatok>2</FkBizonylatok>
  <FkTetelek>2</FkTetelek>
</Ellenorzes>
<Naplok>
  <Naplo>
    <Kod>1</Kod>
    <Nev>OPENING</Nev>
  </Naplo>
  <Naplo>
    <Kod>6</Kod>
    <Nev>Various</Nev>
  </Naplo>
</Naplok>
<Idoszakok>
  <Idoszak>
    <Kod>0</Kod>
    <Nev>OPENING</Nev>
  </Idoszak>
</Idoszakok>
<Szamlaszamok>
  <Szamlaszam>
    <Kod>312</Kod>
    <TKod/>
    <Nev>Customers (foreign)</Nev>
    <ITKod/>
    <INev/>
  </Szamlaszam>
  <Szamlaszam>
    <Kod>413</Kod>
    <TKod/>
    <Nev>Retained earnings</Nev>
    <ITKod/>
    <INev/>
  </Szamlaszam>
  <Szamlaszam>
    <Kod>419</Kod>
    <TKod/>
    <Nev>Profit/loss</Nev>
    <ITKod/>
    <INev/>
  </Szamlaszam>
  <Szamlaszam>
    <Kod>491</Kod>
    <TKod/>
```

```
<Nev>Opening balance</Nev>
<ITKod/>
<INev/>
</Szamlaszam>
</Szamlaszamok>
<Partnerek>
  <Partner>
    <Kod>2001</Kod>
    <TKod/>
    <Nev>DIMITROS GmbH</Nev>
  </Partner>
</Partnerek>
<Rogzitok>
  <Rogzito>
    <Kod>1</Kod>
    <TKod></TKod>
    <Nev>No Data</Nev>
  </Rogzito>
  <Rogzito>
    <Kod>2</Kod>
    <TKod>JDOE</TKod>
    <Nev>John Doe</Nev>
  </Rogzito>
</Rogzitok>
<FkBizonylatok>
  <Biz>
    <BizID>-1</BizID>
    <Naplo>1</Naplo>
    <BizSzam>NYITÁS</BizSzam>
    <Datum>2010-01-01</Datum>
    <Idoszak>0</Idoszak>
  </Biz>
  <Biz>
    <BizID>15</BizID>
    <Naplo>6</Naplo>
    <BizSzam>1</BizSzam>
    <Datum>2010-01-01</Datum>
    <Idoszak>0</Idoszak>
  </Biz>
</FkBizonylatok>
<FkTetelek>
  <Tet>
    <BizID>-1</BizID>
    <TetID>1730</TetID>
    <Szoveg>ANNUAL OPENING</Szoveg>
    <Tartozik>312</Tartozik>
    <Kovetel>491</Kovetel>
    <Osszeg>32500</Osszeg>
    <DevOsszeg>125</DevOsszeg>
    <DevNem>EUR</DevNem>
    <AfaAlap>130000</AfaAlap>
    <AfaKulcs>25</AfaKulcs>
    <Partner>2001</Partner>
```

<PuAzo>2009/00006</PuAzo>
<TeljDatum>2009-08-21</TeljDatum>
<FizHatarido>2009-08-21</FizHatarido>
<Szt/>
<SztTetID/>
<Rogzito>1</Rogzito>
<Rogzitve>2006-05-28 13:54:56</Rogzitve>
</Tet>
<Tet>
<BizID>15</BizID>
<TetID>1824</TetID>
<Szoveg>Adjustment of the profit/loss of the last year</Szoveg>
<Tartozik>419</Tartozik>
<Kovetel>413</Kovetel>
<Osszeg>6157457,44</Osszeg>
<DevOsszeg/>
<DevNem/>
<AfaAlap/>
<AfaKulcs/>
<Partner/>
<PuAzo/>
<TeljDatum/>
<FizHatarido/>
<Szt/>
<SztTetID/>
<Rogzito>2</Rogzito>
<Rogzitve>2006-10-15 10:54:32</Rogzitve>
</Tet>
</FkTetelek>
</Adatok>

3.2 Example of a Ledger Entry with VAT

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<Adatak xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
...
  <Parameterek>
    <SzamlaSzamok>
      <Vevo>311</Vevo>
...
    <BeAFA>467</BeAFA>
  </SzamlaSzamok>
</Parameterek>
...
  <Szamlaszamok>
    <Szamlaszam>
      <Kod>25</Kod>
      <TKod>311</TKod>
      <Nev>Tesztelő Llc.</Nev>
      <ITKod/>
      <INev/>
    </Szamlaszam>
    <Szamlaszam>
      <Kod>114</Kod>
      <TKod>467</TKod>
      <Nev>VAT payable</Nev>
      <ITKod/>
      <INev/>
    </Szamlaszam>
    <Szamlaszam>
      <Kod>163</Kod>
      <TKod>911</TKod>
      <Nev>Domestic Sales revenue</Nev>
      <ITKod/>
      <INev/>
    </Szamlaszam>
  </Szamlaszamok>
...
  <FkBizonylatok>
    <Biz>
      <BizID>1</BizID>
      <Naplo>4</Naplo>
      <BizSzam>FV/08/01-0001</BizSzam>
      <Datum>2008-01-03</Datum>
      <Idoszak>1</Idoszak>
    </Biz>
  </FkBizonylatok>
  <FkTetelek>
    <Tet>
      <BizID>1</BizID>
      <TetID>1</TetID>
      <Szoveg>Sales revenue</Szoveg>
```

```
<Tartozik>25</Tartozik>
<Kovetel>163</Kovetel>
<Osszeg>63000</Osszeg>
<DevOsszeg/>
<DevNem/>
<AfaAlap/>
<AfaKulcs/>
<Partner/>
<PuAzo>2008/001</PuAzo>
<TeljDatum>2008-01-03</TeljDatum>
<FizHatarido>2008-01-18</FizHatarido>
<Szt/>
<SztTetID/>
<Rogzito>1</Rogzito>
<Rogzitve>2008-02-16 10:46:23</Rogzitve>
</Tet>
<Tet>
  <BizID>1</BizID>
  <TetID>2</TetID>
  <Szoveg>Sales revenue</Szoveg>
  <Tartozik>25</Tartozik>
  <Kovetel>114</Kovetel>
  <Osszeg>12600</Osszeg>
  <DevOsszeg/>
  <DevNem/>
  <AfaAlap>63000</AfaAlap>
  <AfaKulcs>20</AfaKulcs>
  <Partner/>
  <PuAzo>2008/001</PuAzo>
  <TeljDatum>2008-01-03</TeljDatum>
  <FizHatarido>2008-01-18</FizHatarido>
  <Szt/>
  <SztTetID/>
  <Rogzito>1</Rogzito>
  <Rogzitve>2008-02-16 10:46:23</Rogzitve>
</Tet>
</FkTetelek>
</Adatok>
```

3.3 Example of a Cancellation Ledger Entry Pair

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<Adatok xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
...
  <Szamlaszamok>
    <Szamlaszam>
      <Kod>36</Kod>
      <TKod>311007</TKod>
      <Nev>Tesztelő Llc.</Nev>
      <ITKod/>
      <INev/>
    </Szamlaszam>
    <Szamlaszam>
      <Kod>59</Kod>
      <TKod>3841</TKod>
      <Nev>Bank account</Nev>
      <ITKod/>
      <INev/>
    </Szamlaszam>
  </Szamlaszamok>
...
  <FkBizonylatok>
    <Biz>
      <BizID>68</BizID>
      <Naplo>2</Naplo>
      <BizSzam>1088</BizSzam>
      <Datum>2009-09-10</Datum>
      <Idoszak>9</Idoszak>
    </Biz>
    <Biz>
      <BizID>113</BizID>
      <Naplo>2</Naplo>
      <BizSzam>1088</BizSzam>
      <Datum>2009-09-10</Datum>
      <Idoszak>9</Idoszak>
    </Biz>
  </FkBizonylatok>
  <FkTetelek>
    <Tet>
      <BizID>68</BizID>
      <TetID>1425</TetID>
      <Szoveg>szla kiegyenlítés</Szoveg>
      <Tartozik>59</Tartozik>
      <Kovetel>36</Kovetel>
      <Osszeg>1250000</Osszeg>
      <DevOsszeg/>
      <DevNem/>
      <AfaAlap/>
      <AfaKulcs/>
      <Partner/>
      <PuAzo>96/2009</PuAzo>
```



```
<TeljDatum/>
<FizHatarido/>
<Szt>I</Szt>
<SztTetID/>
<Rogzito>1</Rogzito>
<Rogzitve>2009-10-09 10:28:15</Rogzitve>
</Tet>
<Tet>
  <BizID>113</BizID>
  <TetID>1524</TetID>
  <Szoveg>ST-szla kiegyenlítés</Szoveg>
  <Tartozik>59</Tartozik>
  <Kovetel>36</Kovetel>
  <Osszeg>-1250000</Osszeg>
  <DevOsszeg/>
  <DevNem/>
  <AfaAlap/>
  <AfaKulcs/>
  <Partner/>
  <PuAzo>96/2009</PuAzo>
  <TeljDatum/>
  <FizHatarido/>
  <Szt/>
  <SztTetID>1425</SztTetID>
  <Rogzito>1</Rogzito>
  <Rogzitve>2009-10-28 10:33:22</Rogzitve>
</Tet>
</FkTetelek>
</Adatok>
```

4 Support for Creating the XML File

4.1 *Technical Suggestions*

When creating the XML file, it is recommended to use the following test cases:

- Use general ledger accounts beginning with “0” in the account numbers segment to avoid errors when identifying the 0111 and 111 accounts. (The <Kod> tag is an integer type, so during parsing of the data the <Kod>0111</Kod> equals to <Kod>111</Kod>, hence the identification number 111 would be duplicated.) Therefore, it is suggested to use a generated number as the value of the <Kod> tag and to give this number as the value of the <Tartozik>, or <Kovetel> tag in the FkTetelek segment.
- Use a name in the partners segment that contains one or more special characters (<, &) to be able to verify that the generated file will be appropriate for the XML parsers.

4.2 *Personal Support*

In case you have any questions regarding the creation of the XML file, contact Gabor Zsigmond of Dimenzió Llc. using the following contact information:

Dimenzió Llc.
Nádorliget St. 8/A.
Budapest, 1117
Tel.: +36-1-311-8804, +36-1-311-0621
Fax: +36-1-473-8805
E-mail: idora@dimenzio-kft.hu

5 Tracking Changes Compared to the Previous Version

5.1 Documentation

Changes compared to the previous version have been indicated with **blue** color in the documentation for easier reading.

5.2 XML Structure

- In the main segments, `<XMLadatok>` tag has been changed to `<XMLAdatok>`, also, `<Rogzítok>` segment has been added.
- In `<XMLAdatok>` segment, version information has been changed to 1.0.1.0 because of the new XML structure.
- `<Ellenorzes>` segment was extended with the `<Rogzítok>` tag.
- `<Szamlaszamok>` segment has been extended with the `<TKod>` tag, the `<IdegenKod>` tag has been changed to `<ITKod>` and the `<IdegenNev>` tag changed to `<INev>`. Also the value of `<ITKod>` has changed from number type to character type.
- `<Partnerek>` segment has been extended with the `<TKod>` segment.
- `<Rogzítok>` segment is included in the main segment now.
- The value type of `<Rogzító>` tag in `<FkTetelek>` has been changed from character type to number type.

6 About XML

The XML (eXtensible Markup Language) file has structured text content, which is readable both by people and the computer. This is why we chose this language.

No special component is needed for the developers to create the file. Just create a file and fill it with content that applies the XML rules to the structure. If the result abides to the XML rules, the content can be viewed in any browser.

For further information about XML, follow the links below:

- Official description: <http://www.w3.org/standards/xml/>
- Wikipedia site: <http://en.wikipedia.org/wiki/XML>

6.1 *Special Characters*

Applications parsing the XML file consider the file faulty if special characters (considering the parser) are included between tags. These characters must be switched before creating the file.

Special character	String to replace it with
<	<
&	&