

XML Interfaces



How to interface external application to WIN€UR

Version 1.0

Le 4 juillet 2007 ADE

XML INTERFACE OVERVIEW 5

WHAT IS IT? 5

WHO IS IT FOR, AND HOW TO IMPLEMENT IT? 5

DOCUMENTS KIT (ZIP FILE) 5

WHAT IS AN XML SCHEMA? 6

HOW TO EXPORT XML DATA FROM WINEUR 6

SAMPLE: 6

HOW TO EXPORT 8

HOW TO IMPORT XML DATA TO WINEUR 10

CREATION OF A DEFINITION FORMAT 10

HOW TO IMPORT A FILE: 16

DETAILED SCHEMA ACCOUNTINGXML.XSD 18

ELEMENT ACCOUNT 19

ELEMENT ADDRESSLINE1 19

ELEMENT ADDRESSLINE2 20

ELEMENT ADDRESSLINE3 20

ELEMENT ADDRESSLINE4 21

ELEMENT ADDRESSLINE5 21

ELEMENT AMOUNTSTOREVALUE 22

ELEMENT ANALYSISPLIT 22

ELEMENT ANALYTICALACCOUNT 22

ELEMENT ANALYTICALAMOUNT 23

ELEMENT ANALYTICALCURRENCY 23

ELEMENT ANALYTICALTYPE 23

ELEMENT BANK 24

ELEMENT BANKACCOUNT 24

ELEMENT BANKADDRESS1 24

ELEMENT BANKADDRESS2 25

ELEMENT BANKADDRESS3 25

ELEMENT BANKADDRESS4 26

ELEMENT BANKADDRESS5 26

ELEMENT BCCYAMOUNT 26

ELEMENT BCCYBALANCE 27

ELEMENT BCCYPAYMENT 27

ELEMENT CLIENTCODE 27

ELEMENT CLIENTCURRENCY 27

ELEMENT CLIENTSUBLEDGERACCOUNT 28

ELEMENT CLIENTVATNUMBER 28



ELEMENT CODE	29
ELEMENT COMMODITY	29
ELEMENT CURRENCY	29
ELEMENT CUSTOMERFLAG	30
ELEMENT DESCRIPTION1	30
ELEMENT DESCRIPTION2	30
ELEMENT DESCRIPTION3	31
ELEMENT DUEDATE	31
ELEMENT EMAIL	31
ELEMENT ENTRYDATE	32
ELEMENT FAX	32
ELEMENT FILENAME	32
ELEMENT FOREX.....	33
ELEMENT GAINSONEXCHANGEACCOUNT	33
ELEMENT INVOICEBANKCODE.....	33
ELEMENT INVOICEDATE	34
ELEMENT INVOICEDUEDATE	34
ELEMENT INVOICENUMBER.....	34
ELEMENT INVOICETEXT.....	35
ELEMENT JOURNALCODE	35
ELEMENT LANGUAGE	35
ELEMENT LOSSESONEXCHANGEACCOUNT	36
ELEMENT MASTERANALYTICALACCOUNT.....	36
ELEMENT NAME	36
ELEMENT NETMASSKG.....	37
ELEMENT NUMBER	37
ELEMENT OCCYAMOUNT	37
ELEMENT OCCYBALANCE	38
ELEMENT OCCYPAYMENT	38
ELEMENT PAYMENT	38
ELEMENT PAYMENTDATE.....	38
ELEMENT PHONE.....	39
ELEMENT PRINTCODE	39
ELEMENT SUBLEDGERACCOUNT	39
ELEMENT SUBTYPE	40
ELEMENT TAXCODE	40
ELEMENT TEXT	41
ELEMENT TEXTLINE1.....	41
ELEMENT TEXTLINE2.....	41
ELEMENT TRANSCORBANKACCOUNT	42
ELEMENT TYPE	42
ELEMENT TYPEOFINVOICE.....	43
ELEMENT VALUEDATE	43
ELEMENT VATNUMBER	43

ELEMENT VATNUMBERUSED BYTRANSCOR	44
ELEMENT VERSION.....	44
ELEMENT VOUCHER	45
ELEMENT WEBSITE	45
ELEMENT XML2WINEUR	45
ELEMENT XML2WINEUR/GENERALLEDGERACCOUNTS	47
ELEMENT XML2WINEUR/ANALYTICALACCOUNTS	47
ELEMENT XML2WINEUR/CLIENTS.....	47
ELEMENT XML2WINEUR/INVOICES	48
ELEMENT XML2WINEUR/ENTRIES	48
COMPLEXTYPE ADDRESSESTYPE	49
ELEMENT ADDRESSESTYPE/ADDRESS.....	49
COMPLEXTYPE ADDRESSTYPE	50
COMPLEXTYPE ANALYTICALACCOUNTSTYPE	50
ELEMENT ANALYTICALACCOUNTSTYPE/ANALYTICALACCOUNT	51
COMPLEXTYPE ANALYTICALACCOUNTTYPE	52
COMPLEXTYPE CLIENTLINETYPE	53
ELEMENT CLIENTLINETYPE/INVOICEVATINFORMATION	56
COMPLEXTYPE CLIENTSTYPE.....	56
ELEMENT CLIENTSTYPE/CLIENT	57
COMPLEXTYPE CLIENTTYPE	59
ELEMENT CLIENTTYPE/ADDRESSES	62
COMPLEXTYPE ENTRIESTYPE	63
ELEMENT ENTRIESTYPE/ENTRY	63
COMPLEXTYPE ENTRYTYPE.....	64
ELEMENT ENTRYTYPE/HEADER	64
ELEMENT ENTRYTYPE/GENERALLEDGERLINE.....	65
ELEMENT ENTRYTYPE/CLIENTLINE.....	66
COMPLEXTYPE GENERALLEDGERACCOUNTSTYPE	68
ELEMENT GENERALLEDGERACCOUNTSTYPE/GENERALLEDGERACCOUNT	68
COMPLEXTYPE GENERALLEDGERACCOUNTTYPE	69
COMPLEXTYPE GENERALLEDGERLINETYPE	71
COMPLEXTYPE HEADERTYPE	72
COMPLEXTYPE INVOICESTYPE	73
ELEMENT INVOICESTYPE/INVOICE	73
COMPLEXTYPE INVOICETYPE	75
COMPLEXTYPE INVOICEVATINFORMATIONTYPE	77

XML INTERFACE OVERVIEW

WHAT IS IT?

The XML Interface is a set of standards which will enable you to understand how to create XML files from your application to be uploaded to WIN€UR. The same XML Interface is used to download data from WIN€UR to your application.

For instance, when you are uploading an XML file for an accounting transaction, the fields of the accounting entry will be automatically filled up using the data within the XML file.

- XML stands for EXtensible Markup Language
- XML is a markup language much like HTML
- XML was designed to describe data
- XML tags are not predefined. You must define your own tags
- XML uses a Document Type Definition (DTD) or an XML Schema to describe the data
- XML with a DTD or XML Schema is designed to be self-descriptive
- XML is a W3C Recommendation

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
<to>Dove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

WHO IS IT FOR, AND HOW TO IMPLEMENT IT?

In a typical implementation, you will need to engage a contractor or acquire the assistance of your organisation's internal IT Department to create a programmatic interface with existing information system.

This interface will then create the necessary XML files needed for the upload or the download. Such an interface would alleviate the problem of re-keying of information from an existing accounting information system to a invoicing program. The benefit of such an interface is essentially the time it saves, by reducing the work required in filling out the data fields.

DOCUMENTS KIT (ZIP FILE)

The XML documentation kit is contained on a compressed zip file.

This file contains the following:

This master document
AccountingXml.xsd
AccountingXml.dtd

This is the schemas file
This is the DTD file

WHAT IS AN XML SCHEMA?

A newer XML schema language, described by the W3C as the successor of DTDs, is XML Schema, or more informally referred to by the initialism for XML Schema instances, XSD (XML Schema Definition).

XSDs are far more powerful than DTDs in describing XML languages.

They use a rich datatyping system, allow for more detailed constraints on an XML document's logical structure, and must be processed in a more robust validation framework. XSDs also use an XML-based format which makes it possible to use ordinary XML tools to help process them, although XSD implementations require much more than just the ability to read XML.

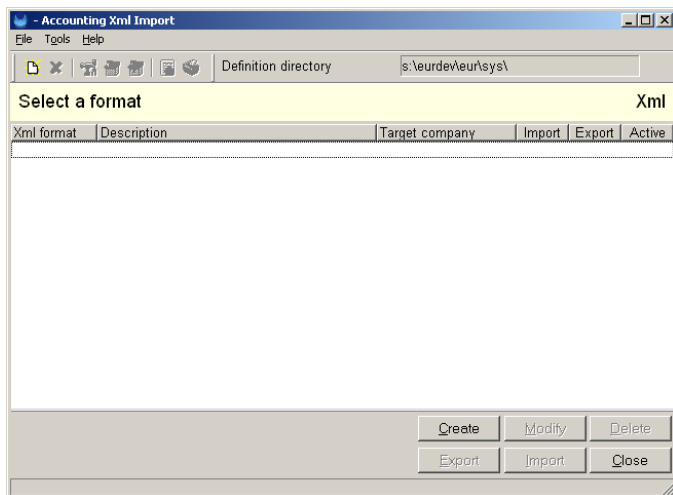
HOW TO EXPORT XML DATA FROM WINEUR

SAMPLE:

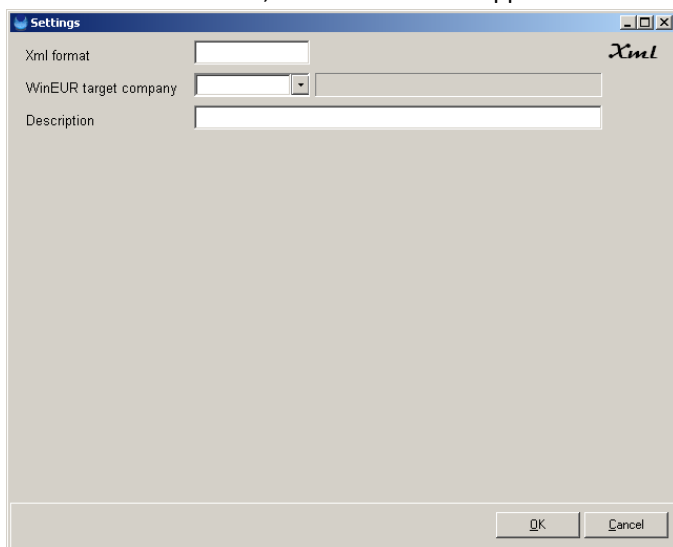
In order to understand the XML file syntax, the easiest way is to create an export from accounting data of an existing company and to look at the result.

Creation of a definition format

Launch the program from the menu Import /Export XML



Press the Create button, This screen will be appears

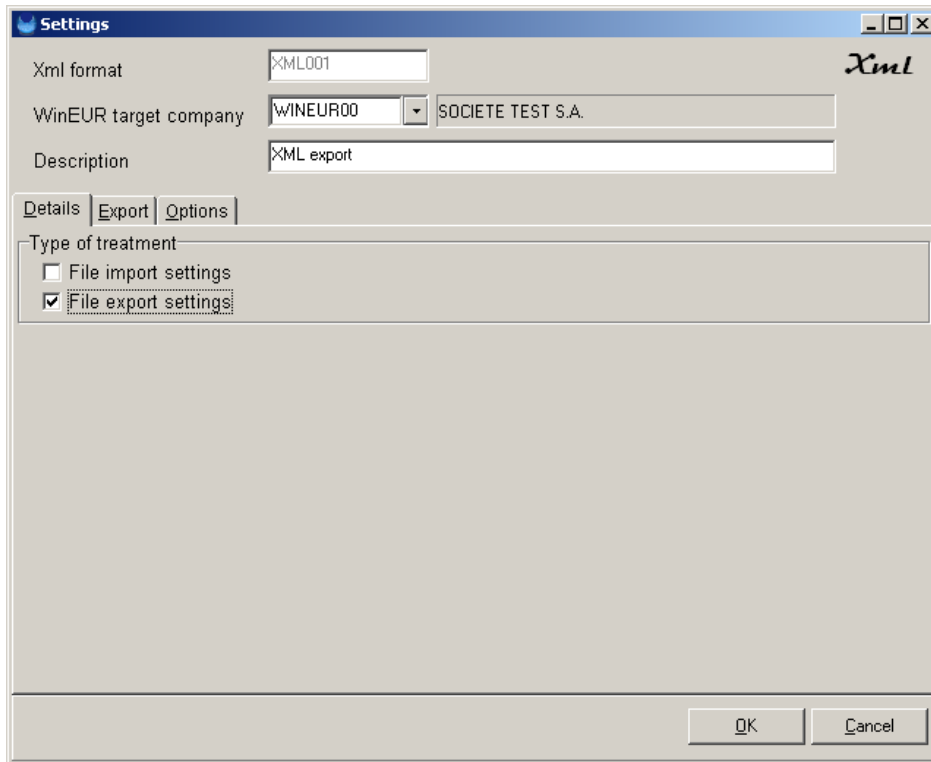


Xml format : File name of the export format.

WinEUR target company : Select the WinEUR company code from which data will be exported.

Description : File description

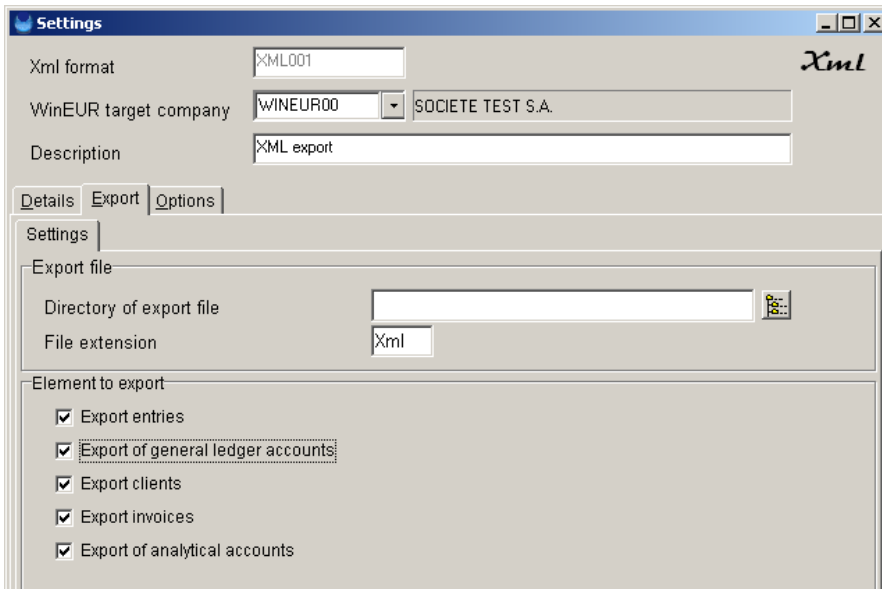
After entering the above information, the following screen will appear :



The screenshot shows a 'Settings' dialog box with the following fields and options:

- Xml format:** A text box containing 'XML001' and an 'Xml' icon to the right.
- WinEUR target company:** A dropdown menu showing 'WINEUR00' and a text box containing 'SOCIETE TEST S.A.'.
- Description:** A text box containing 'XML export'.
- Details | Export | Options:** A tabbed interface with 'Details' selected.
- Type of treatment:** A section containing two checkboxes:
 - File import settings
 - File export settings
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

Select File export settings
 Select the Export tab



Directory of export file : Define the directory in which the file will be saved (By default, in the company directory)

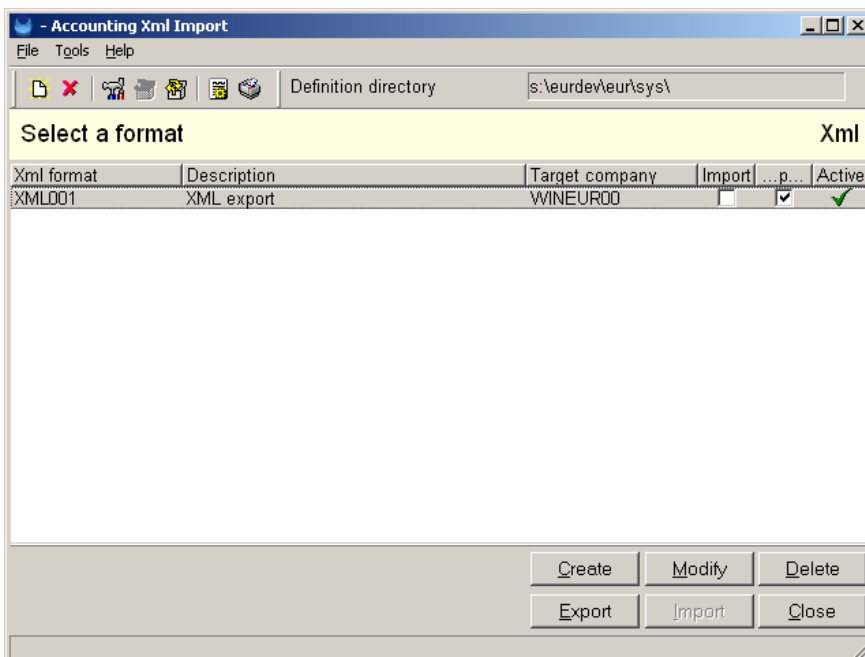
Element to export : Click data that you wish to export. (entries, GL accounts, clients, etc..)

Information in « Options » tab are correctly configured by default

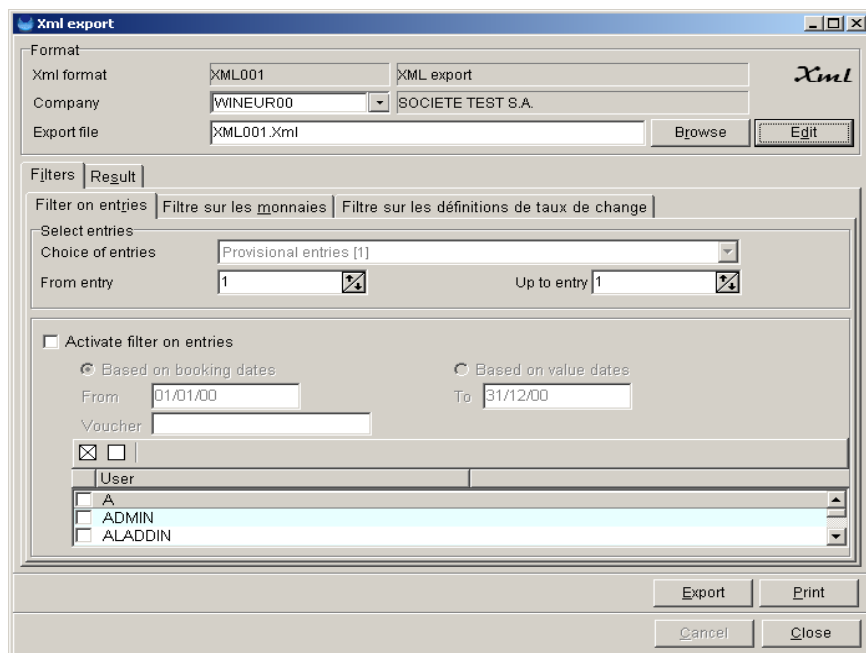
Click OK to save the file definition

HOW TO EXPORT A FILE

In the main menu, choose the definition file you wish to use



Click then on the Export button.
The following screen appears :



The screenshot shows a dialog box titled "xml export". It has a "Format" section with fields for "Xml format" (XML001), "Company" (WINEUR00), and "Export file" (XML001.xml). There are "Browse" and "Edit" buttons. Below this is a "Filters" section with tabs for "Filter on entries", "Filtre sur les monnaies", and "Filtre sur les définitions de taux de change". The "Filter on entries" tab is active, showing "Choice of entries" (Provisional entries [1]), "From entry" (1), and "Up to entry" (1). There are checkboxes for "Activate filter on entries", "Based on booking dates", and "Based on value dates". The "Based on booking dates" section has "From" (01/01/00) and "To" (31/12/00) fields. A "Voucher" field is also present. At the bottom, there is a "User" list with checkboxes for "A", "ADMIN", and "ALADDIN". The "ADMIN" user is selected. At the very bottom of the dialog are "Export", "Print", "Cancel", and "Close" buttons.

By default, all entries will be saved. If you wish to make a selection (entries, currencies, etc..) indicate your choice using the corresponding fields.

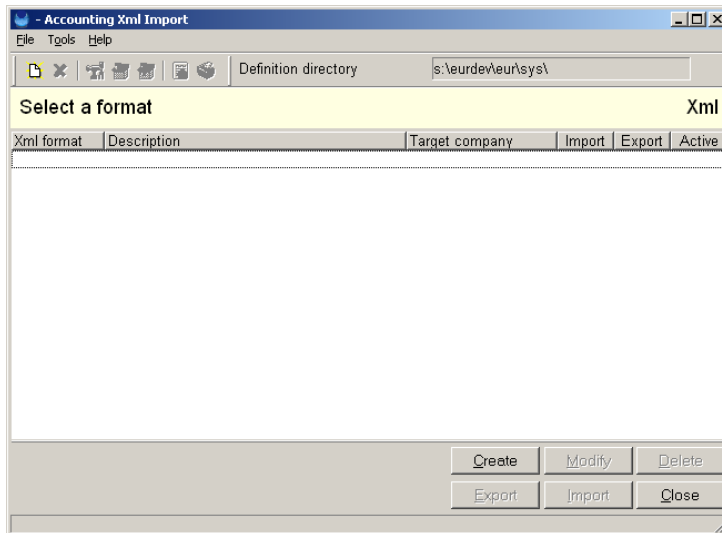
Then click on Export button to launch the treatment.

In the result tab, the program will indicate treatments which are executed.

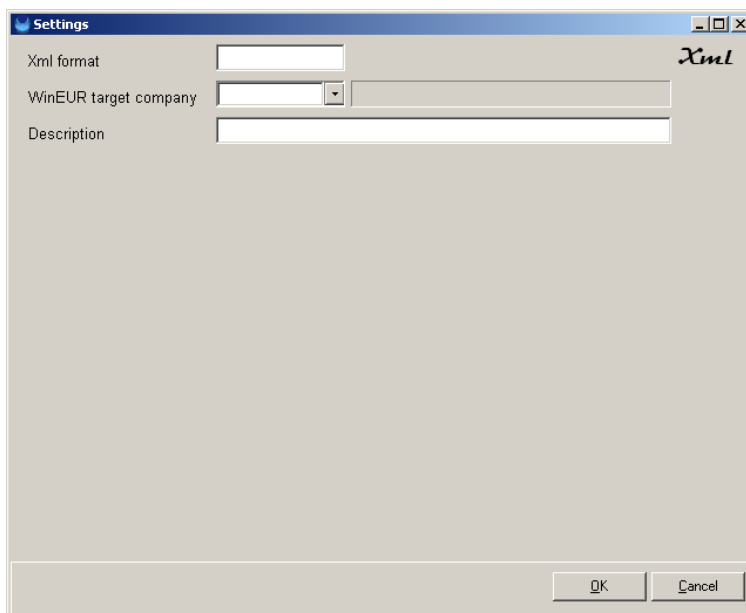
At the end of the treatment, you can visualize your file by clicking on the Edit button. (top left).

CREATION OF A DEFINITION FORMAT

Launch the program from the Import/Export XML



Press the Create button, This screen will appear

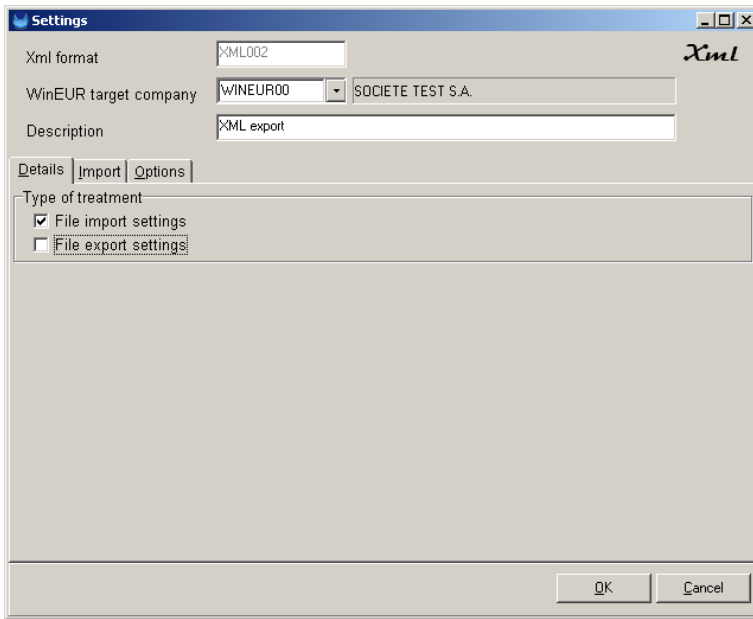


Xml format : File name of the import format.

WinEUR target company : Select the WinEUR company code in which data will be imported.

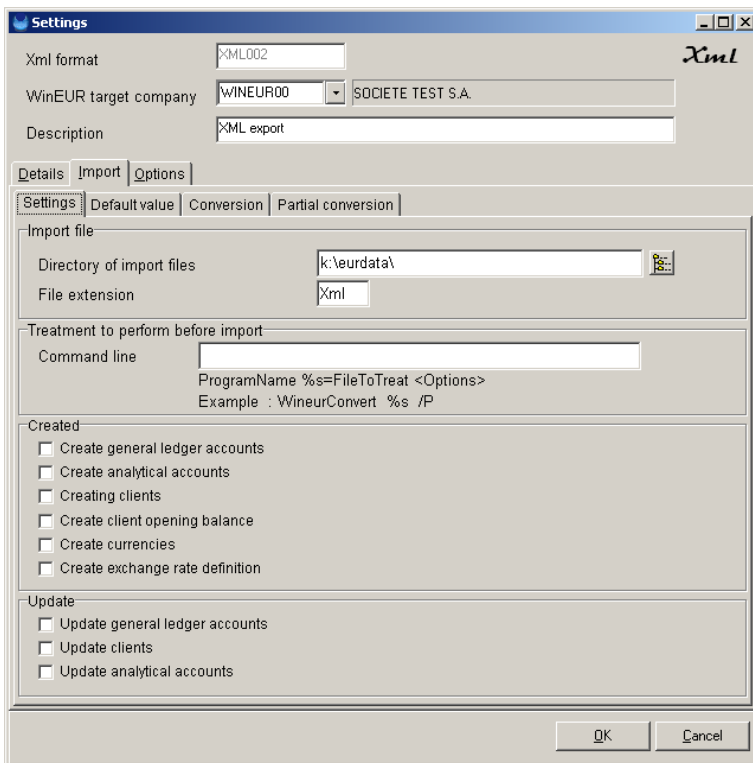
Description : File description

Then click OK. You will have now to define the import settings. You should select the format and click on modify. Then click on file import settings and the following screen will appear :



The screenshot shows the 'Settings' dialog box with the 'Details' tab selected. The 'Xml format' is set to 'XML002'. The 'WinEUR target company' is 'WINEUR00' and 'SOCIETE TEST S.A.'. The 'Description' is 'XML export'. Under 'Type of treatment', 'File import settings' is checked and 'File export settings' is unchecked. 'OK' and 'Cancel' buttons are at the bottom.

Select the import tab



The screenshot shows the 'Settings' dialog box with the 'Import' tab selected. The 'Import file' section has 'Directory of import files' set to 'k:\eurdata\' and 'File extension' set to 'Xml'. The 'Treatment to perform before import' section has a 'Command line' field with the text 'ProgramName %s=FileToTreat <Options>' and an example 'Example : WineurConvert %s /P'. There are two sections of checkboxes: 'Created' (Create general ledger accounts, Create analytical accounts, Creating clients, Create client opening balance, Create currencies, Create exchange rate definition) and 'Update' (Update general ledger accounts, Update clients, Update analytical accounts). 'OK' and 'Cancel' buttons are at the bottom.

In the **Settings** tab, you will find all informations concerning import

Directory of import file : Define the directory from which you will read the file to be imported.

File extension : define the file extension. (xml by default).

Treatment to perform before import : please do not consider this part as it as been done only for a specific customer.

Created

Tick actions which should be done during import

Update

Tick actions which should be done during import

Select Default value tab

Be aware that default values are used only for creation and not for update

During imports, it can happen that information are nonexistent. For instance when you import a client, the language code does not exist in the program from which you import. It means then you can “force” to have a language code GB

Default value for Entry

- Default currency code : self explaining
- Multiple tax code : self explaining
- General ledger debit line text : self explaining
- General ledger credit line text : self explaining
- Client debit line text : self explaining
- Client credit line text : self explaining

Default value for Account

Settings		Default value	Conversion	Partial conversion
Entry	Account	Clients	Analytical account	
Revaluation	Not evaluated			
Amount to evaluate	Balance			
Gains on exchange account			?	
Losses on exchange account			?	
Analysis split	No analytical bookings		<input type="checkbox"/> Update general ledger	

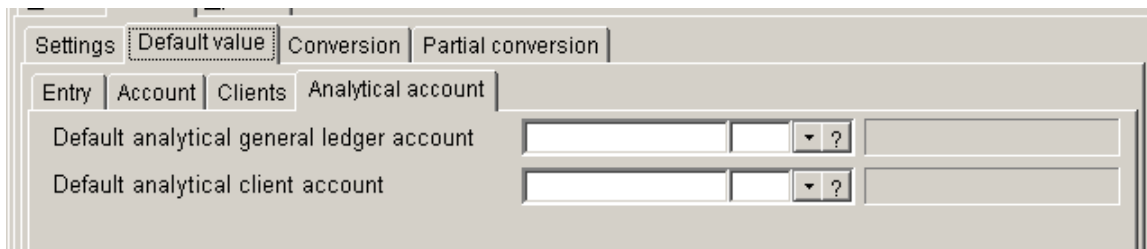
Revaluation : Revaluation method you wish to use.
 Amount to evaluate : which amount has to be evaluated
 Gains on exchange account : self explaining
 Losses on exchange account : self explaining
 Analysis split : define the analytic breakdown
 Update general ledger : Tick if you wish to update the existing GL accounts.

Default value for Clients

Settings		Default value	Conversion	Partial conversion
Entry	Account	Clients	Analytical account	
Client sub-ledger account			?	
Supplier sub-ledger account			?	
Language code				
Payment term				
Default tax code				
Revaluation	Not evaluated			
Amount to evaluate	Balance			
Gains on exchange account			?	
Losses on exchange account			?	

Client sub-ledger account : self explaining
 Supplier sub-ledger account : self explaining
 Language code : self explaining
 Payment term : self explaining
 Default tax code : self explaining
 Revaluation : Revaluation method you wish to use.
 Amount to evaluate : which amount has to be evaluated
 Gains on exchange account : self explaining
 Losses on exchange account : self explaining

Default value for Analytical account

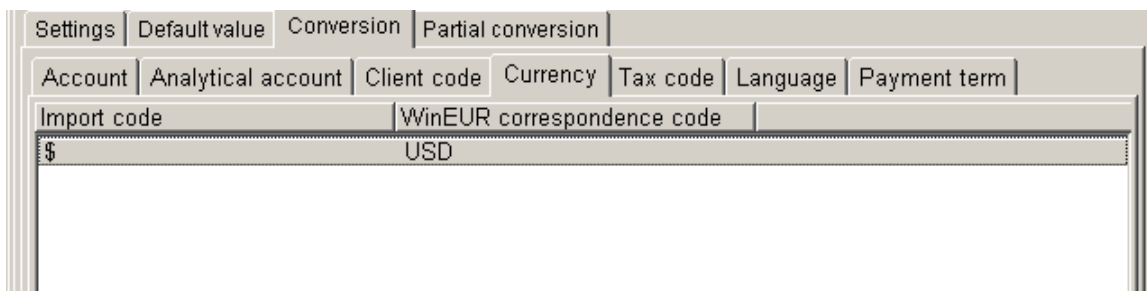


Default analytical general ledger account : self explaining
 Default analytical client account : self explaining.

Conversion tab

During import, it can happen that codes are different between two applications. Therefore you have to define conversion between two codes. For instance in the application from which you wish to import the currency code is \$ while it is USD in WinEUR.

You should use this tab to define conversions.



Import code	WinEUR correspondence code
\$	USD

You can establish conversions for the following fields :

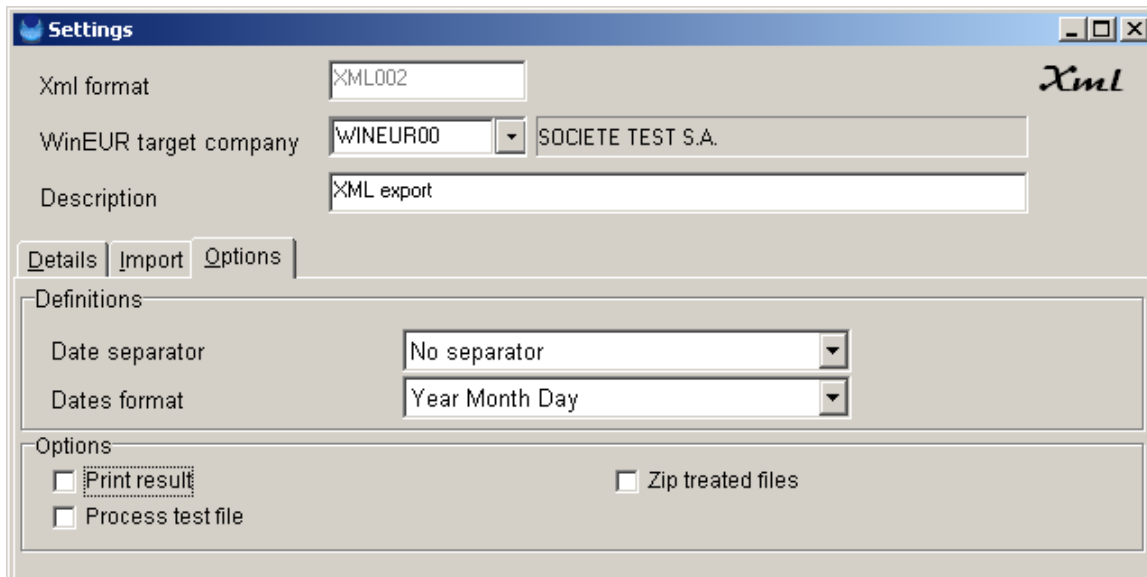
Account : GL account,
 Anytical account : self explaining,
 Client code : debtors and creditors code
 Currency : self explaining
 Tax code : self explaining
 Language : self explaining
 Payment term : self explaining

Partial conversion tab

Allows to execute accounts partial conversion.

If you wish to choose this type of conversion, please contact to get more informations.

Options tab



The screenshot shows a 'Settings' dialog box with the 'Options' tab selected. The 'Details' tab is also visible. The 'Options' section contains three checkboxes: 'Print result', 'Process test file', and 'Zip treated files'. The 'Definitions' section contains two dropdown menus: 'Date separator' (set to 'No separator') and 'Dates format' (set to 'Year Month Day'). The 'WinEUR target company' dropdown is set to 'WINEUR00' and the text field next to it contains 'SOCIETE TEST S.A.'. The 'Description' field contains 'XML export'. The 'Xml format' field contains 'XML002'. The 'Xml' logo is visible in the top right corner of the dialog box.

Définitions :

Date separator and dates format : should not be used as there is a standard XML format by default.

Options

Print result : tick if you wish to print import result during automatic treatment.

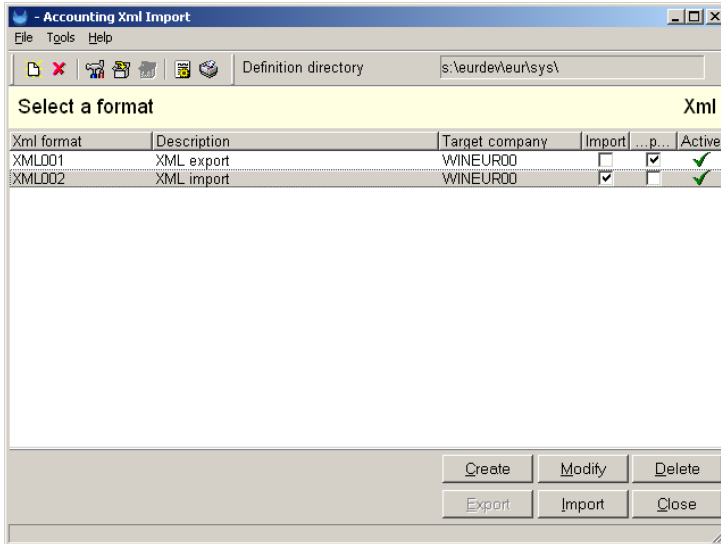
Zip treated files : After automatic treatment, the program will save the XMLfile in the import directory under the form of a zip file. (File name : <year><month>.zip)

Process test file : if you tick this box it will only test the file without importing data.

Click OK to save the import definition.

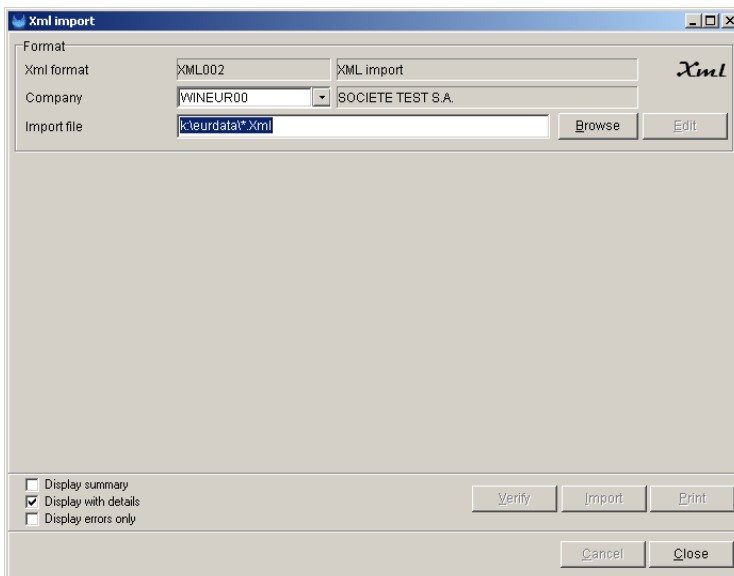
HOW TO IMPORT A FILE:

In the main menu, choose the definition file you wish to use.

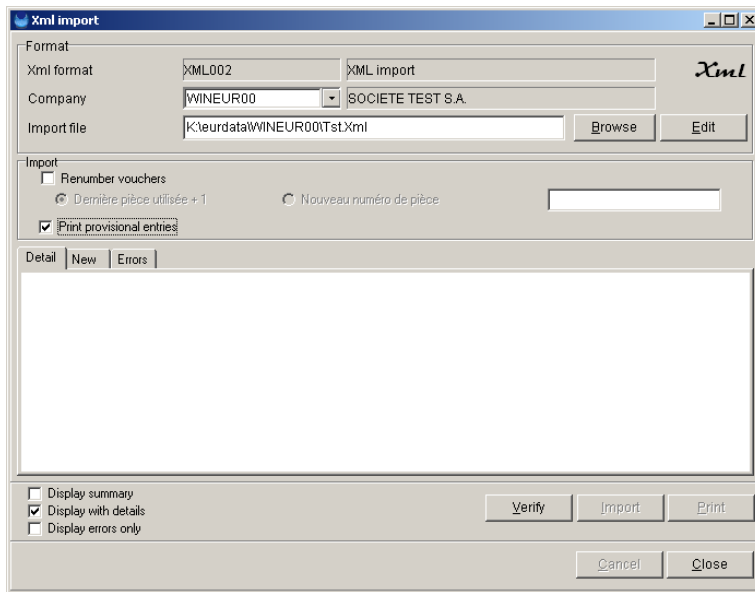


Click on the import button.

The following screen will appear :



Click on « Browse » to select the file, then the following screen will appear :



You can visualize the file when clicking on Edit.

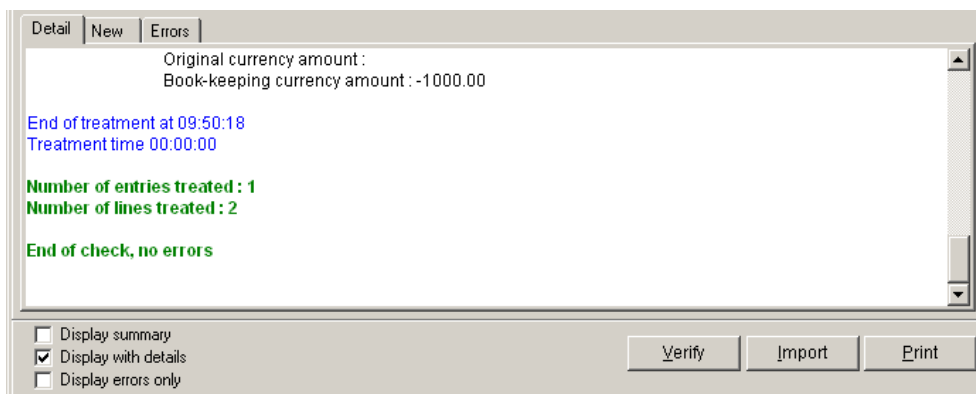
If necessary, complete informations on Import.

Then click on the verify button.

The program will execute a pre-treatment to verify the data coherence

If there is an error, the program will display the exact problem precising for instance that there is an unknown currency. You will have then to create the new currency in WinEUR or execute a conversion.

Be aware that as long all errors are not corrected, you will not be able to import data and the import button is inaccessible.



If there is no error, the message : End of check, no errors will appear at the end of verification.

The import button is now accessible and you can then import data.

Please note that you can see in the "new" tab all new elements that have been created while you are going to see only errors in the "errors" tab..

Once the file has been treated, it will be renamed (zipped if the option has been ticked) in order not to import it several times in a row.

DETAILED SCHEMA ACCOUNTINGXML.XSD

Elements

Account
 AddressLine1
 AddressLine2
 AddressLine3
 AddressLine4
 AddressLine5
 AmountStoreValue
 AnalysisSplit
 AnalyticalAccount
 AnalyticalAmount
 AnalyticalCurrency
 AnalyticalType
 Bank
 BankAccount
 BankAddress1
 BankAddress2
 BankAddress3
 BankAddress4
 BankAddress5
 BccyAmount
 BccyBalance
 BccyPayment
 ClientCode
 ClientCurrency
 ClientSubledgerAccount
 ClientVATNumber
 Code
 Commodity
 Currency
 CustomerFlag
 Description1
 Description2
 Description3
 DueDate
 Email
 EntryDate
 Fax
 FileName
 Forex
 GainsOnExchangeAccount
 InvoiceBankCode
 InvoiceDate
 InvoiceDuedate
 InvoiceNumber
 InvoiceText
 JournalCode
 Language
 LossesOnExchangeAccount
 MasterAnalyticalAccount

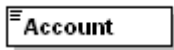
Complex types

AddressesType
 AddressType
 AnalyticalAccountsType
 AnalyticalAccountType
 ClientLineType
 ClientsType
 ClientType
 EntriesType
 EntryType
 GeneralLedgerAccountsType
 GeneralLedgerAccountType
 GeneralLedgerLineType
 HeaderType
 InvoicesType
 InvoiceType
 InvoiceVATInformationType

Name
 NetMassKg
 Number
 OccyAmount
 OccyBalance
 OccyPayment
 Payment
 PaymentDate
 Phone
 PrintCode
 SubledgerAccount
 SubType
 TaxCode
 Text
 TextLine1
 TextLine2
 TranscorBankAccount
 Type
 TypeOfInvoice
 ValueDate
 VatNumber
 VATNumberUsedByTranscor
 Version
 Voucher
 WebSite
 Xml2WinEUR

ELEMENT ACCOUNT

diagram



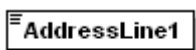
Account number

type restriction of xs:string
 properties content simple
 used by complexTypes [GeneralLedgerAccountType](#) [GeneralLedgerLineType](#)
 facets maxLength 15
 annotation documentation
 Account number
 source

```
<xs:element name="Account">
  <xs:annotation>
    <xs:documentation>Account number</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="15"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

ELEMENT ADDRESSLINE1


diagram



Client Address line 1

type	restriction of xs:string
properties	content simple
used by	complexType AddressType
facets	maxLength 40
annotation	documentation Client Address line 1
source	<pre> <xs:element name="AddressLine1"> <xs:annotation> <xs:documentation>Client Address line 1</xs:documentation> </xs:annotation> <xs:simpleType> <xs:annotation> <xs:documentation>Client address line 1</xs:documentation> </xs:annotation> <xs:restriction base="xs:string"> <xs:maxLength value="40"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>

ELEMENT ADDRESSLINE2

diagram	 <p>Client Address line 2</p>
type	restriction of xs:string
properties	content simple
used by	complexType AddressType
facets	maxLength 40
annotation	documentation Client Address line 2
source	<pre> <xs:element name="AddressLine2"> <xs:annotation> <xs:documentation>Client Address line 2</xs:documentation> </xs:annotation> <xs:simpleType> <xs:annotation> <xs:documentation>Client address line 2</xs:documentation> </xs:annotation> <xs:restriction base="xs:string"> <xs:maxLength value="40"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>

ELEMENT ADDRESSLINE3

diagram	 <p>Client Address line 3</p>
type	restriction of xs:string
properties	content simple
used by	complexType AddressType
facets	maxLength 40
annotation	documentation Client Address line 3
source	<pre> <xs:element name="AddressLine3"> <xs:annotation> <xs:documentation>Client Address line 3</xs:documentation> </xs:annotation> <xs:simpleType> <xs:annotation> </pre>

```

<xs:documentation>Client address line 3</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
  <xs:maxLength value="40"/>
</xs:restriction>
</xs:simpleType>
</xs:element>

```

ELEMENT ADDRESSLINE4

diagram



Client Address line 4

type restriction of xs:string
 properties content simple
 used by complexType [AddressType](#)
 facets maxLength 40
 annotation documentation

Client Address line 4

source

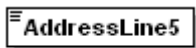
```

<xs:element name="AddressLine4">
  <xs:annotation>
    <xs:documentation>Client Address line 4</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:annotation>
      <xs:documentation>Client address line 4</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:maxLength value="40"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

```

ELEMENT ADDRESSLINE5

diagram



Client Address line 5

type restriction of xs:string
 properties content simple
 used by complexType [AddressType](#)
 facets maxLength 40
 annotation documentation

Client Address line 5

source

```

<xs:element name="AddressLine5">
  <xs:annotation>
    <xs:documentation>Client Address line 5</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:annotation>
      <xs:documentation>Client address line 5</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:maxLength value="40"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

```

ELEMENT AMOUNTSTOREVALUE

diagram

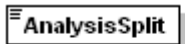


Revalued amounts.

type	extension of xs:boolean
properties	content complex
used by	complexType ClientType GeneralLedgerAccountType
annotation	documentation Revalued amounts.
source	<pre> <xs:element name="AmountStoreValue"> <xs:annotation> <xs:documentation>Revalued amounts.</xs:documentation> </xs:annotation> <xs:complexType> <xs:simpleContent> <xs:extension base="xs:boolean"/> </xs:simpleContent> </xs:complexType> </xs:element> </pre>

ELEMENT ANALYSISPLIT

diagram



Contain the rule of analysis split. (ex Obligatory, optional, no split)

type	restriction of xs:string
properties	content simple
used by	complexType GeneralLedgerAccountType
facets	enumeration NoAnalyticalBookings enumeration ObligatorySplit
annotation	documentation Contain the rule of analysis split. (ex Obligatory, optional, no split)
source	<pre> <xs:element name="AnalysisSplit"> <xs:annotation> <xs:documentation>Contain the rule of analysis split. (ex Obligatory, optional, no split)</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="NoAnalyticalBookings"/> <xs:enumeration value="ObligatorySplit"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>

ELEMENT ANALYTICALACCOUNT

diagram



Analytical account

type	restriction of xs:string
properties	content simple
used by	complexType AnalyticalAccountType ClientLineType GeneralLedgerLineType InvoiceType
facets	maxLength 15
annotation	documentation Analytical account
source	<pre> <xs:element name="AnalyticalAccount"> <xs:annotation> <xs:documentation>Analytical account</xs:documentation> </xs:annotation> <xs:simpleType> </pre>

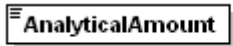
```

<xs:restriction base="xs:string">
  <xs:maxLength value="15"/>
</xs:restriction>
</xs:simpleType>
</xs:element>

```

ELEMENT ANALYTICALAMOUNT

diagram



Analytical amount in analytical currency

type	xs:decimal
properties	content simple
used by	complexType ClientLineType GeneralLedgerLineType
annotation	documentation Analytical amount in analytical currency
source	<pre> <xs:element name="AnalyticalAmount" type="xs:decimal"> <xs:annotation> <xs:documentation>Analytical amount in analytical currency</xs:documentation> </xs:annotation> </xs:element> </pre>

ELEMENT ANALYTICALCURRENCY

diagram

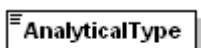


Iso currency code

type	restriction of xs:string
properties	content simple
used by	complexType ClientLineType GeneralLedgerLineType
facets	maxLength 3
annotation	documentation Iso currency code
source	<pre> <xs:element name="AnalyticalCurrency"> <xs:annotation> <xs:documentation>Iso currency code</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="3"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>

ELEMENT ANALYTICALTYPE

diagram



Specific field

type	restriction of xs:string
properties	content simple
used by	complexType AnalyticalAccountType
facets	enumeration Detail enumeration Master
annotation	documentation Specific field
source	<pre> <xs:element name="AnalyticalType"> <xs:annotation> <xs:documentation>Specific field</xs:documentation> </pre>

```

</xs:annotation>
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:enumeration value="Detail"/>
    <xs:enumeration value="Master"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>

```

ELEMENT BANK

diagram

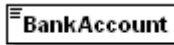


Client bank

type	restriction of xs:string
properties	content simple
used by	complexType ClientType
facets	maxLength 25
annotation	documentation Client bank
source	<pre> <xs:element name="Bank"> <xs:annotation> <xs:documentation>Client bank</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="25"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>

ELEMENT BANKACCOUNT

diagram

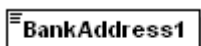


Client bank account

type	restriction of xs:string
properties	content simple
used by	complexType ClientType
facets	maxLength 25
annotation	documentation Client bank account
source	<pre> <xs:element name="BankAccount"> <xs:annotation> <xs:documentation>Client bank account</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="25"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>

ELEMENT BANKADDRESS1

diagram



Client bank address line 1

type	restriction of xs:string
------	--------------------------

properties	content	simple
used by	complexType	ClientType
facets	maxLength	40
annotation	documentation	Client bank address line 1
source	<pre> <xs:element name="BankAddress1"> <xs:annotation> <xs:documentation>Client bank address line 1</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="40"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>	

ELEMENT BANKADDRESS2

diagram	 <p>Client bank address line 2</p>	
type	restriction of xs:string	
properties	content	simple
used by	complexType	ClientType
facets	maxLength	40
annotation	documentation	Client bank address line 2
source	<pre> <xs:element name="BankAddress2"> <xs:annotation> <xs:documentation>Client bank address line 2</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="40"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>	

ELEMENT BANKADDRESS3

diagram	 <p>Client bank address line 3</p>	
type	restriction of xs:string	
properties	content	simple
used by	complexType	ClientType
facets	maxLength	40
annotation	documentation	Client bank address line 3
source	<pre> <xs:element name="BankAddress3"> <xs:annotation> <xs:documentation>Client bank address line 3</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="40"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>	

ELEMENT BANKADDRESS4

diagram

BankAddress4

Client bank address line 4

type	restriction of xs:string
properties	content simple
used by	complexType ClientType
facets	maxLength 40
annotation	documentation Client bank address line 4
source	<pre> <xs:element name="BankAddress4"> <xs:annotation> <xs:documentation>Client bank address line 4</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="40"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>

ELEMENT BANKADDRESS5

diagram

BankAddress5

Client bank address line 5

type	restriction of xs:string
properties	content simple
used by	complexType ClientType
facets	maxLength 60
annotation	documentation Client bank address line 5
source	<pre> <xs:element name="BankAddress5"> <xs:annotation> <xs:documentation>Client bank address line 5</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="60"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>

ELEMENT BCCYAMOUNT

diagram

BccyAmount

Amount in book-keeping currency

type	xs:decimal
properties	content simple
used by	complexTypees ClientLineType GeneralLedgerLineType InvoiceType
annotation	documentation Amount in book-keeping currency
source	<pre> <xs:element name="BccyAmount" type="xs:decimal"> <xs:annotation> <xs:documentation>Amount in book-keeping currency</xs:documentation> </xs:annotation> </xs:element> </pre>

ELEMENT BCCYBALANCE

diagram

BccyBalance

Balance in book-keeping currency

type **xs:decimal**
 properties content simple
 used by complexType [InvoiceType](#)
 annotation documentation
 documentation Balance in book-keeping currency
 source `<xs:element name="BccyBalance" type="xs:decimal">`
`<xs:annotation>`
`<xs:documentation>Balance in book-keeping currency</xs:documentation>`
`</xs:annotation>`
`</xs:element>`

ELEMENT BCCYPAYMENT

diagram

BccyPayment

Payment in book-keeping currency

type **xs:decimal**
 properties content simple
 used by complexType [InvoiceType](#)
 annotation documentation
 documentation Payment in book-keeping currency
 source `<xs:element name="BccyPayment" type="xs:decimal">`
`<xs:annotation>`
`<xs:documentation>Payment in book-keeping currency</xs:documentation>`
`</xs:annotation>`
`</xs:element>`

ELEMENT CLIENTCODE

diagram

ClientCode

Client code

type restriction of **xs:string**
 properties content simple
 used by complexTypes [ClientLineType](#) [InvoiceType](#)
 facets maxLength 15
 annotation documentation
 documentation Client code
 source `<xs:element name="ClientCode">`
`<xs:annotation>`
`<xs:documentation>Client code</xs:documentation>`
`</xs:annotation>`
`<xs:simpleType>`
`<xs:restriction base="xs:string">`
`<xs:maxLength value="15"/>`
`</xs:restriction>`
`</xs:simpleType>`
`</xs:element>`

ELEMENT CLIENTCURRENCY

diagram

ClientCurrency

Iso currency code

type **xs:string**
 properties content simple
 used by complexType [ClientLineType](#)
 annotation documentation
 Iso currency code
 source

```
<xs:element name="ClientCurrency" type="xs:string">
  <xs:annotation>
    <xs:documentation>Iso currency code</xs:documentation>
  </xs:annotation>
</xs:element>
```

ELEMENT CLIENTSUBLEDGERACCOUNT

diagram

ClientSubledgerAccount

Sub-ledger control account

type **restriction of xs:string**
 properties content simple
 used by complexType [ClientLineType](#)
 facets maxLength 15
 annotation documentation
 Sub-ledger control account
 source

```
<xs:element name="ClientSubledgerAccount">
  <xs:annotation>
    <xs:documentation>Sub-ledger control account</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="15"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

ELEMENT CLIENTVATNUMBER

diagram

ClientVATNumber

Not used

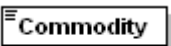
type **restriction of xs:string**
 properties content simple
 used by complexType [InvoiceVATInformationType](#)
 facets maxLength 15
 annotation documentation
 Not used
 source

```
<xs:element name="ClientVATNumber">
  <xs:annotation>
    <xs:documentation>Not used</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="15"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

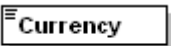
ELEMENT CODE

diagram	 Client code
type	restriction of xs:string
properties	content simple
used by	complexType ClientType
facets	maxLength 15
annotation	documentation Client code
source	<pre> <xs:element name="Code"> <xs:annotation> <xs:documentation>Client code</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="15"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>

ELEMENT COMMODITY

diagram	 Not used
type	restriction of xs:string
properties	content simple
used by	complexType InvoiceVATInformationType
facets	maxLength 15
annotation	documentation Not used
source	<pre> <xs:element name="Commodity"> <xs:annotation> <xs:documentation>Not used</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="15"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>

ELEMENT CURRENCY

diagram	 Iso currency code
type	restriction of xs:string
properties	content simple
used by	complexTypees AnalyticalAccountType ClientType GeneralLedgerAccountType GeneralLedgerLineType InvoiceType
facets	length 3
annotation	documentation Iso currency code
source	<pre> <xs:element name="Currency"> <xs:annotation> <xs:documentation>Iso currency code</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> </pre>

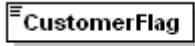
```

<xs:length value="3"/>
</xs:restriction>
</xs:simpleType>
</xs:element>

```

ELEMENT CUSTOMERFLAG

diagram



Yes = customer / No = supplier

type	restriction of xs:string
properties	content simple
used by	complexType ClientType
facets	enumeration no
	enumeration yes
annotation	documentation
	Yes = customer / No = supplier
source	<pre> <xs:element name="CustomerFlag"> <xs:annotation> <xs:documentation>Yes = customer / No = supplier</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="no"/> <xs:enumeration value="yes"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>

ELEMENT DESCRIPTION1

diagram

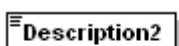


Analytical description

type	restriction of xs:string
properties	content simple
used by	complexType AnalyticalAccountType
facets	maxLength 40
annotation	documentation
	Analytical description
source	<pre> <xs:element name="Description1"> <xs:annotation> <xs:documentation>Analytical description</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="40"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>

ELEMENT DESCRIPTION2

diagram

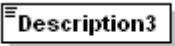


Analytical description line 2


type	restriction of xs:string
properties	content simple
used by	complexType AnalyticalAccountType

facets	maxLength 40
annotation	documentation Analytical description line 2
source	<pre> <xs:element name="Description2"> <xs:annotation> <xs:documentation>Analytical description line 2</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="40"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>

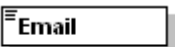
ELEMENT DESCRIPTION3

diagram	 <p>Analytical description line 3</p>
type	restriction of xs:string
properties	content simple
used by	complexType AnalyticalAccountType
facets	maxLength 40
annotation	documentation Analytical description line 3
source	<pre> <xs:element name="Description3"> <xs:annotation> <xs:documentation>Analytical description line 3</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="40"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>

ELEMENT DUEDATE


diagram	 <p>Invoice due date</p>
type	xs:date
properties	content simple
used by	complexType InvoiceType
annotation	documentation Invoice due date
source	<pre> <xs:element name="DueDate" type="xs:date"> <xs:annotation> <xs:documentation>Invoice due date</xs:documentation> </xs:annotation> </xs:element> </pre>

ELEMENT EMAIL


diagram	 <p>Client email</p>
type	restriction of xs:string
properties	content simple

used by complexType [ClientType](#)
 facets maxLength 60
 annotation documentation
 Client email
 source `<xs:element name="Email">
 <xs:annotation>
 <xs:documentation>Client email</xs:documentation>
 </xs:annotation>
 <xs:simpleType>
 <xs:restriction base="xs:string">
 <xs:maxLength value="60"/>
 </xs:restriction>
 </xs:simpleType>
 </xs:element>`

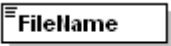
ELEMENT ENTRYDATE

diagram 
 Entry date
 type xs:date
 properties content simple
 used by complexType [HeaderType](#)
 annotation documentation
 Entry date
 source `<xs:element name="EntryDate" type="xs:date">
 <xs:annotation>
 <xs:documentation>Entry date</xs:documentation>
 </xs:annotation>
 </xs:element>`

ELEMENT FAX

diagram 
 Client faxnumber
 type restriction of xs:string
 properties content simple
 used by complexType [ClientType](#)
 facets maxLength 25
 annotation documentation
 Client faxnumber
 source `<xs:element name="Fax">
 <xs:annotation>
 <xs:documentation>Client faxnumber</xs:documentation>
 </xs:annotation>
 <xs:simpleType>
 <xs:restriction base="xs:string">
 <xs:maxLength value="25"/>
 </xs:restriction>
 </xs:simpleType>
 </xs:element>`

ELEMENT FILENAME

diagram 
 Xml file name
 type xs:string


properties used by annotation source

content element documentation
simple
[Xml2WinEUR](#)
 Xml file name

```
<xs:element name="FileName" type="xs:string">
  <xs:annotation>
    <xs:documentation>Xml file name</xs:documentation>
  </xs:annotation>
</xs:element>
```

ELEMENT FOREX

diagram



Revaluation of exchange rates.


type extension of xs:boolean
 properties used by annotation source

content complex
 complexTypes [ClientType](#) [GeneralLedgerAccountType](#)
 documentation
 Revaluation of exchange rates.

```
<xs:element name="Forex">
  <xs:annotation>
    <xs:documentation>Revaluation of exchange rates.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:boolean"/>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

ELEMENT GAINSONEXCHANGEACCOUNT

diagram



Federal ledger account : Gains on exchange

type restriction of xs:string
 properties used by facets annotation source

content simple
 complexTypes [ClientType](#) [GeneralLedgerAccountType](#)
 maxLength 15
 documentation
 Federal ledger account : Gains on exchange

```
<xs:element name="GainsOnExchangeAccount">
  <xs:annotation>
    <xs:documentation>Federal ledger account : Gains on exchange</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="15"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

ELEMENT INVOICEBANKCODE

diagram



Invoice bank code

type restriction of xs:string

properties content simple
 used by complexType [ClientLineType](#)
 facets maxLength 15
 annotation documentation
 Invoice bank code
 source `<xs:element name="InvoiceBankCode">
 <xs:annotation>
 <xs:documentation>Invoice bank code</xs:documentation>
 </xs:annotation>
 <xs:simpleType>
 <xs:restriction base="xs:string">
 <xs:maxLength value="15"/>
 </xs:restriction>
 </xs:simpleType>
 </xs:element>`

ELEMENT INVOICEDATE

diagram

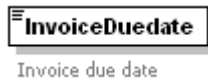


Invoice date

type xs:date
 properties content simple
 used by complexTypes [ClientLineType](#) [InvoiceType](#)
 annotation documentation
 Invoice date
 source `<xs:element name="InvoiceDate" type="xs:date">
 <xs:annotation>
 <xs:documentation>Invoice date</xs:documentation>
 </xs:annotation>
 </xs:element>`

ELEMENT INVOICEDUEDATE

diagram

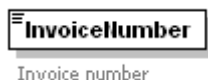


Invoice due date

type xs:date
 properties content simple
 used by complexType [ClientLineType](#)
 annotation documentation
 Invoice due date
 source `<xs:element name="InvoiceDuedate" type="xs:date">
 <xs:annotation>
 <xs:documentation>Invoice due date</xs:documentation>
 </xs:annotation>
 </xs:element>`

ELEMENT INVOICENUMBER

diagram



Invoice number

type xs:string
 properties content simple
 used by complexType [ClientLineType](#)
 annotation documentation
 Invoice number
 source `<xs:element name="InvoiceNumber" type="xs:string">`

```

<xs:annotation>
  <xs:documentation>Invoice number</xs:documentation>
</xs:annotation>
</xs:element>

```

ELEMENT INVOICETEXT

diagram



Invoice text

type restriction of xs:string
 properties content simple
 used by complexType [ClientLineType](#)
 facets maxLength 30
 annotation documentation

Invoice text

source

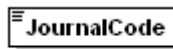
```

<xs:element name="InvoiceText">
  <xs:annotation>
    <xs:documentation>Invoice text</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="30"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

```

ELEMENT JOURNALCODE

diagram



Journal code

type xs:string
 properties content simple
 used by complexType [HeaderType](#)
 annotation documentation

Journal code

source

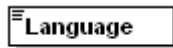
```

<xs:element name="JournalCode" type="xs:string">
  <xs:annotation>
    <xs:documentation>Journal code</xs:documentation>
  </xs:annotation>
</xs:element>

```

ELEMENT LANGUAGE

diagram



Client language

type restriction of xs:string
 properties content simple
 used by complexType [ClientType](#)
 facets maxLength 3
 annotation documentation

Client language

source

```

<xs:element name="Language">
  <xs:annotation>
    <xs:documentation>Client language</xs:documentation>
  </xs:annotation>
  <xs:simpleType>

```

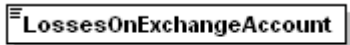
```

<xs:restriction base="xs:string">
  <xs:maxLength value="3"/>
</xs:restriction>
</xs:simpleType>
</xs:element>

```

ELEMENT LOSSESONEXCHANGEACCOUNT

diagram



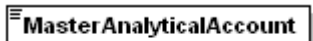
LossesOnExchangeAccount

General ledger account : Losses on exchange

type	restriction of xs:string
properties	content simple
used by	complexType ClientType GeneralLedgerAccountType
facets	maxLength 15
annotation	documentation General ledger account : Losses on exchange
source	<pre> <xs:element name="LossesOnExchangeAccount"> <xs:annotation> <xs:documentation>General ledger account : Losses on exchange</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="15"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>

ELEMENT MASTERANALYTICALACCOUNT

diagram



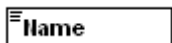
MasterAnalyticalAccount

Specific

type	restriction of xs:string
properties	content simple
used by	complexType AnalyticalAccountType
facets	maxLength 15
annotation	documentation Specific
source	<pre> <xs:element name="MasterAnalyticalAccount"> <xs:annotation> <xs:documentation>Specific</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="15"/> </xs:restriction> </xs:simpleType> </xs:element> </pre>

ELEMENT NAME

diagram



Name


Client name

type	restriction of xs:string
properties	content simple
used by	complexType ClientType
facets	maxLength 30

annotation documentation
 Client name

source `<xs:element name="Name">`
`<xs:annotation>`
`<xs:documentation>Client name</xs:documentation>`
`</xs:annotation>`
`<xs:simpleType>`
`<xs:restriction base="xs:string">`
`<xs:maxLength value="30"/>`
`</xs:restriction>`
`</xs:simpleType>`
`</xs:element>`

ELEMENT NETMASSKG


diagram 

Not used

type `xs:decimal`
 properties content simple
 used by complexType [InvoiceVATInformationType](#)
 annotation documentation
 Not used

source `<xs:element name="NetMassKg" type="xs:decimal">`
`<xs:annotation>`
`<xs:documentation>Not used</xs:documentation>`
`</xs:annotation>`
`</xs:element>`

ELEMENT NUMBER


diagram 

Invoice number

type `xs:string`
 properties content simple
 used by complexType [InvoiceType](#)
 annotation documentation
 Invoice number

source `<xs:element name="Number" type="xs:string">`
`<xs:annotation>`
`<xs:documentation>Invoice number</xs:documentation>`
`</xs:annotation>`
`</xs:element>`

ELEMENT OCCYAMOUNT

diagram 

Amount in original currency

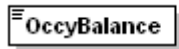
type `xs:decimal`
 properties content simple
 used by complexTypes [ClientLineType](#) [GeneralLedgerLineType](#) [InvoiceType](#)
 annotation documentation
 Amount in original currency

source `<xs:element name="OccyAmount" type="xs:decimal">`
`<xs:annotation>`
`<xs:documentation>Amount in original currency</xs:documentation>`
`</xs:annotation>`

</xs:element>

ELEMENT OCCYBALANCE

diagram



Invoice balance Amount in original currency

type **xs:decimal**

properties content simple

used by complexType [InvoiceType](#)

annotation documentation

Invoice balance Amount in original currency

source `<xs:element name="OccyBalance" type="xs:decimal">`

`<xs:annotation>`

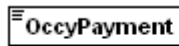
`<xs:documentation>Invoice balance Amount in original currency</xs:documentation>`

`</xs:annotation>`

`</xs:element>`

ELEMENT OCCYPAYMENT

diagram



Invoice payment Amount in original currency

type **xs:decimal**

properties content simple

used by complexType [InvoiceType](#)

annotation documentation

Invoice payment Amount in original currency

source `<xs:element name="OccyPayment" type="xs:decimal">`

`<xs:annotation>`

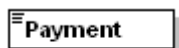
`<xs:documentation>Invoice payment Amount in original currency</xs:documentation>`

`</xs:annotation>`

`</xs:element>`

ELEMENT PAYMENT

diagram



Payment terms

type **xs:string**

properties content simple

used by complexType [ClientType](#)

annotation documentation

Payment terms

source `<xs:element name="Payment" type="xs:string">`

`<xs:annotation>`

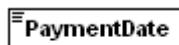
`<xs:documentation>Payment terms</xs:documentation>`

`</xs:annotation>`

`</xs:element>`

ELEMENT PAYMENTDATE

diagram



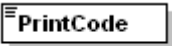
Invoice payment date

type	xs:date
properties	content simple
used by	complexType InvoiceType
annotation	documentation Invoice payment date
source	<pre><xs:element name="PaymentDate" type="xs:date"> <xs:annotation> <xs:documentation>Invoice payment date</xs:documentation> </xs:annotation> </xs:element></pre>

ELEMENT PHONE

diagram	 <p>Client phone number</p>
type	restriction of xs:string
properties	content simple
used by	complexType ClientType
facets	maxLength 25
annotation	documentation Client phone number
source	<pre><xs:element name="Phone"> <xs:annotation> <xs:documentation>Client phone number</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="25"/> </xs:restriction> </xs:simpleType> </xs:element></pre>

ELEMENT PRINTCODE

diagram	 <p>Analytical Print code</p>
type	restriction of xs:string
properties	content simple
used by	complexType AnalyticalAccountType
facets	maxLength 3
annotation	documentation Analytical Print code
source	<pre><xs:element name="PrintCode"> <xs:annotation> <xs:documentation>Analytical Print code</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="3"/> </xs:restriction> </xs:simpleType> </xs:element></pre>

ELEMENT SUBLEDGERACCOUNT

diagram

SubledgerAccount

Client sub-ledger control account

type
properties
used by
facets
annotation

restriction of xs:string
content simple
complexType [ClientType](#)
maxLength 15
documentation
Client sub-ledger control account

source

```
<xs:element name="SubledgerAccount">
  <xs:annotation>
    <xs:documentation>Client sub-ledger control account</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="15"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

ELEMENT SUBTYPE

diagram

SubType

Account sub type

type
properties
used by
facets

restriction of xs:string
content simple
complexType [GeneralLedgerAccountType](#)
enumeration NormalAccount
enumeration Sub-LedgerControlAccount
enumeration InventoryAccount
enumeration Class
enumeration Group
enumeration SubClass
enumeration SubGroup
enumeration SecuritiesAccount

annotation

documentation

Account sub type

source

```
<xs:element name="SubType">
  <xs:annotation>
    <xs:documentation>Account sub type</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="NormalAccount"/>
      <xs:enumeration value="Sub-LedgerControlAccount"/>
      <xs:enumeration value="InventoryAccount"/>
      <xs:enumeration value="Class"/>
      <xs:enumeration value="Group"/>
      <xs:enumeration value="SubClass"/>
      <xs:enumeration value="SubGroup"/>
      <xs:enumeration value="SecuritiesAccount"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

ELEMENT TAXCODE

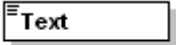
diagram

TaxCode

Tax code

type	xs:string
properties	content simple
used by	complexType ClientLineType ClientType InvoiceType
annotation	documentation
	Tax code
source	<pre><xs:element name="TaxCode" type="xs:string"> <xs:annotation> <xs:documentation>Tax code</xs:documentation> </xs:annotation> </xs:element></pre>

ELEMENT TEXT

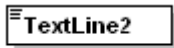
diagram	 <p>Account description</p>
type	restriction of xs:string
properties	content simple
used by	complexType GeneralLedgerAccountType InvoiceType
facets	maxLength 30
annotation	documentation
	Account description
source	<pre><xs:element name="Text"> <xs:annotation> <xs:documentation>Account description</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="30"/> </xs:restriction> </xs:simpleType> </xs:element></pre>

ELEMENT TEXTLINE1

diagram	 <p>Entry line 1</p>
type	restriction of xs:string
properties	content simple
used by	complexType ClientLineType GeneralLedgerLineType
facets	maxLength 30
annotation	documentation
	Entry line 1
source	<pre><xs:element name="TextLine1"> <xs:annotation> <xs:documentation>Entry line 1</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="30"/> </xs:restriction> </xs:simpleType> </xs:element></pre>

ELEMENT TEXTLINE2

diagram



Entry line 2

type restriction of xs:string
 properties content simple
 used by complexTypes [ClientLineType](#) [GeneralLedgerLineType](#)
 facets maxLength 30
 annotation documentation
 Entry line 2
 source

```
<xs:element name="TextLine2">
  <xs:annotation>
    <xs:documentation>Entry line 2</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="30"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

ELEMENT TRANSCORBANKACCOUNT

diagram



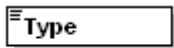
Not used

type restriction of xs:string
 properties content simple
 used by complexType [InvoiceVATInformationType](#)
 facets maxLength 15
 annotation documentation
 Not used
 source

```
<xs:element name="TranscorBankAccount">
  <xs:annotation>
    <xs:documentation>Not used</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="15"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

ELEMENT TYPE

diagram



Type

type restriction of xs:string
 properties content simple
 used by complexTypes [ClientLineType](#) [GeneralLedgerAccountType](#) [GeneralLedgerLineType](#)
 facets enumeration DEBIT
 enumeration CREDIT
 annotation documentation
 Type
 source

```
<xs:element name="Type">
  <xs:annotation>
    <xs:documentation>Type </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="DEBIT"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

```

<xs:enumeration value="CREDIT"/>
</xs:restriction>
</xs:simpleType>
</xs:element>

```

ELEMENT TYPEOFINVOICE

diagram

TypeOfInvoice

Invoice type (ex Supplier invoice, customer credit note...)

type	restriction of xs:string
properties	content simple
used by	complexType InvoiceType
facets	enumeration CustomerCreditNote enumeration PurchaseInvoice enumeration SalesInvoice enumeration SupplierCreditNote enumeration SupplierDebitNote enumeration CustomerDebitNote
annotation	documentation Invoice type (ex Supplier invoice, customer credit note...)

```

source
<xs:element name="TypeOfInvoice">
  <xs:annotation>
    <xs:documentation>Invoice type (ex Supplier invoice, customer credit note...)</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="CustomerCreditNote"/>
      <xs:enumeration value="PurchaseInvoice"/>
      <xs:enumeration value="SalesInvoice"/>
      <xs:enumeration value="SupplierCreditNote"/>
      <xs:enumeration value="SupplierDebitNote"/>
      <xs:enumeration value="CustomerDebitNote"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

```

ELEMENT VALUEDATE

diagram

ValueDate

Value date

type	xs:date
properties	content simple
used by	complexType HeaderType
annotation	documentation Value date

```

source
<xs:element name="ValueDate" type="xs:date">
  <xs:annotation>
    <xs:documentation>Value date</xs:documentation>
  </xs:annotation>
</xs:element>

```

ELEMENT VATNUMBER

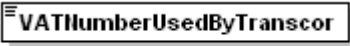
diagram

VatNumber

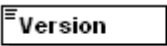
Client VAT number

type	restriction of xs:string
properties	content simple
used by	complexType ClientType
facets	maxLength 15
annotation	documentation Client VAT number
source	<pre><xs:element name="VatNumber"> <xs:annotation> <xs:documentation>Client VAT number</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="15"/> </xs:restriction> </xs:simpleType> </xs:element></pre>

ELEMENT VATNUMBERUSEDBYTRANSCOR

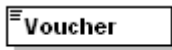
diagram	 <p>Not used</p>
type	restriction of xs:string
properties	content simple
used by	complexType InvoiceVATInformationType
facets	maxLength 15
annotation	documentation Not used
source	<pre><xs:element name="VATNumberUsedByTranscor"> <xs:annotation> <xs:documentation>Not used</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength value="15"/> </xs:restriction> </xs:simpleType> </xs:element></pre>

ELEMENT VERSION

diagram	 <p>Format version : constant should be A001</p>
type	restriction of xs:string
properties	content simple
used by	element Xmi2WinEUR
facets	enumeration A001
annotation	documentation Format version : constant should be A001
source	<pre><xs:element name="Version"> <xs:annotation> <xs:documentation>Format version : constant should be A001</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="A001"/> </xs:restriction> </xs:simpleType> </xs:element></pre>

ELEMENT VOUCHER

diagram



Entry voucher

type
properties
used by
annotation

xs:string
content simple
complexType [HeaderType](#)
documentation

source

```
<xs:element name="Voucher" type="xs:string">
  <xs:annotation>
    <xs:documentation>Entry voucher</xs:documentation>
  </xs:annotation>
</xs:element>
```

ELEMENT WEBSITE

diagram



Client Web site

type
properties
used by
facets
annotation

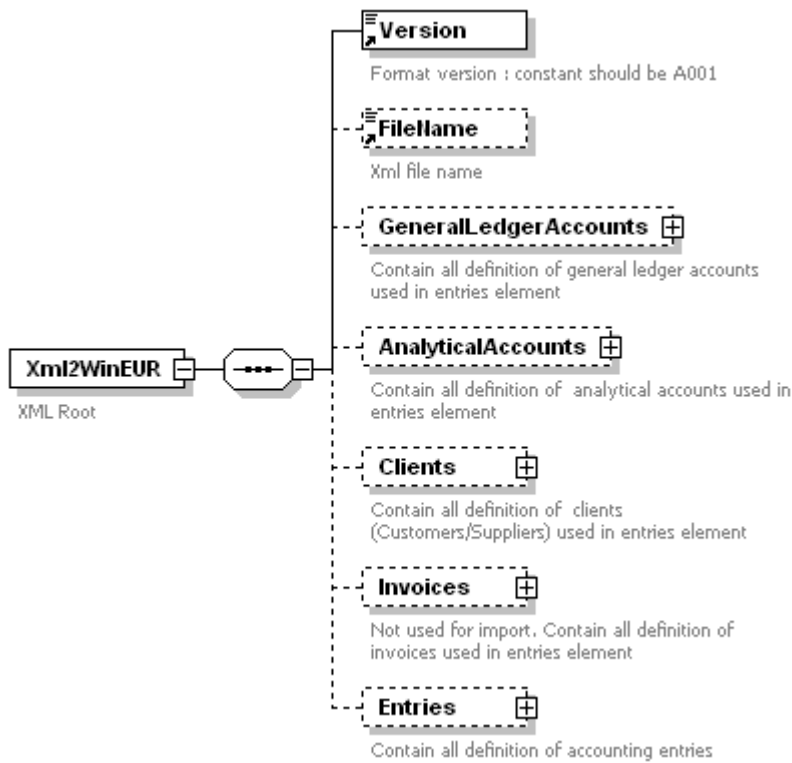
restriction of xs:string
content simple
complexType [ClientType](#)
maxLength 60
documentation

source

```
<xs:element name="WebSite">
  <xs:annotation>
    <xs:documentation>Client Web site</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="60"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

ELEMENT XML2WINEUR

diagram



properties
children
annotation
source

```

content      complex
Version FileName GeneralLedgerAccounts AnalyticalAccounts Clients Invoices Entries
documentation
XML Root
<xs:element name="Xml2WinEUR">
  <xs:annotation>
    <xs:documentation>XML Root</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Version">
        <xs:annotation>
          <xs:documentation>Format version : constant should be A001</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element ref="FileName" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Xml file name</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="GeneralLedgerAccounts" type="GeneralLedgerAccountsType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Contain all definition of general ledger accounts used in entries element</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="AnalyticalAccounts" type="AnalyticalAccountsType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Contain all definition of analytical accounts used in entries element</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Clients" type="ClientsType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Contain all definition of clients (Customers/Suppliers) used in entries element</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Invoices" type="InvoicesType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Not used for import. Contain all definition of invoices used in entries element</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
  
```

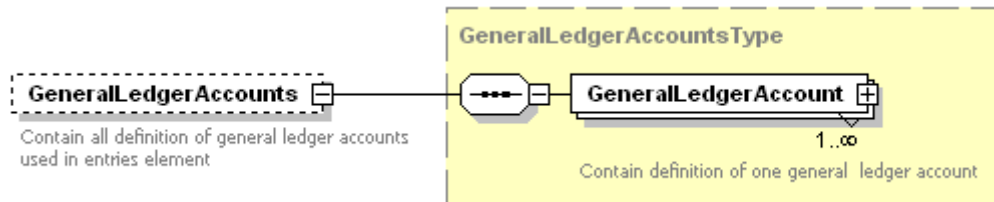
```

</xs:element>
<xs:element name="Entries" type="EntriesType" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Contain all definition of accounting entries</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

```

ELEMENT XML2WINEUR/GENERALLEDGERACCOUNTS

diagram



type [GeneralLedgerAccountsType](#)

properties
 isRef 0
 minOccurs 0
 maxOccurs 1
 content complex

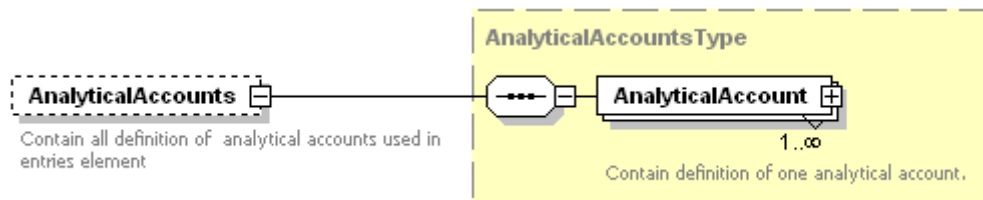
children [GeneralLedgerAccount](#)

annotation
 documentation
 Contain all definition of general ledger accounts used in entries element

source
 <xs:element name="GeneralLedgerAccounts" type="GeneralLedgerAccountsType" minOccurs="0">
 <xs:annotation>
 <xs:documentation>Contain all definition of general ledger accounts used in entries element</xs:documentation>
 </xs:annotation>
 </xs:element>

ELEMENT XML2WINEUR/ANALYTICALACCOUNTS

diagram



type [AnalyticalAccountsType](#)

properties
 isRef 0
 minOccurs 0
 maxOccurs 1
 content complex

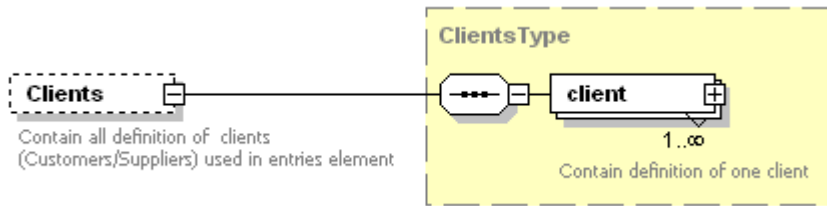
children [AnalyticalAccount](#)

annotation
 documentation
 Contain all definition of analytical accounts used in entries element

source
 <xs:element name="AnalyticalAccounts" type="AnalyticalAccountsType" minOccurs="0">
 <xs:annotation>
 <xs:documentation>Contain all definition of analytical accounts used in entries element</xs:documentation>
 </xs:annotation>
 </xs:element>

ELEMENT XML2WINEUR/CLIENTS

diagram



type
properties

ClientsType
isRef 0
minOcc 0
maxOcc 1
content complex

children
annotation

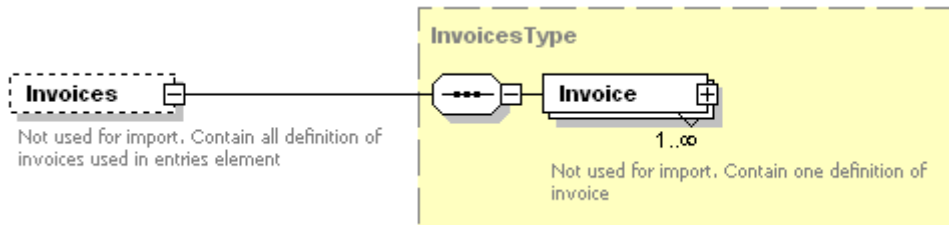
client
documentation
Contain all definition of clients (Customers/Suppliers) used in entries element

source

```
<xs:element name="Clients" type="ClientsType" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Contain all definition of clients (Customers/Suppliers) used in entries element</xs:documentation>
  </xs:annotation>
</xs:element>
```

ELEMENT XML2WINEUR/INVOICES

diagram



type
properties

InvoicesType
isRef 0
minOcc 0
maxOcc 1
content complex

children
annotation

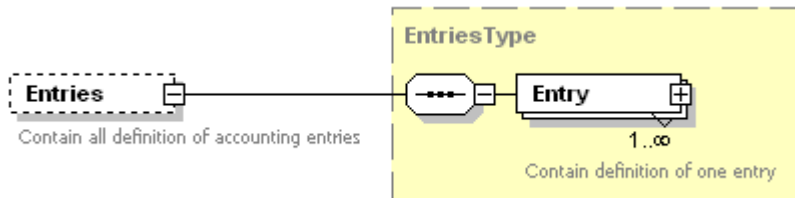
Invoice
documentation
Not used for import. Contain all definition of invoices used in entries element

source

```
<xs:element name="Invoices" type="InvoicesType" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Not used for import. Contain all definition of invoices used in entries element</xs:documentation>
  </xs:annotation>
</xs:element>
```

ELEMENT XML2WINEUR/ENTRIES

diagram



type
properties

EntriesType
isRef 0
minOcc 0
maxOcc 1
content complex

children
annotation
source

Entry

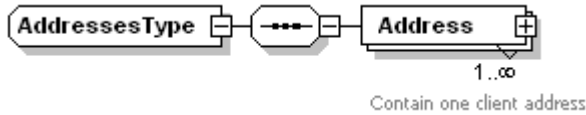
documentation

Contain all definition of accounting entries

```
<xs:element name="Entries" type="EntriesType" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Contain all definition of accounting entries</xs:documentation>
  </xs:annotation>
</xs:element>
```

COMPLEXTYPE ADRESSESTYPE

diagram



children
used by
source

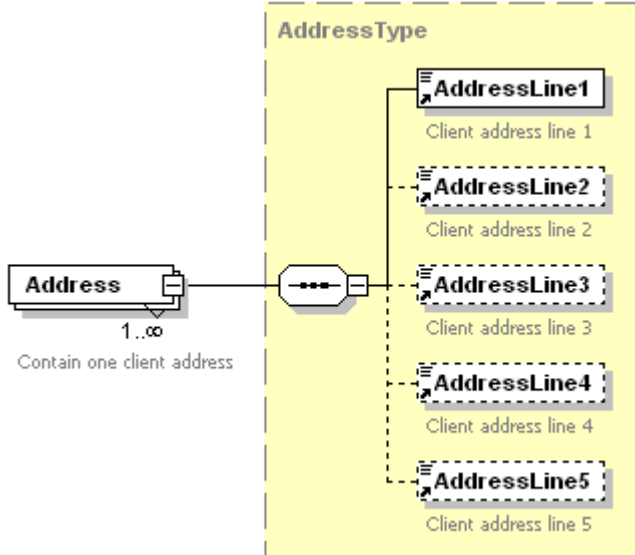
Address

element [ClientType/Addresses](#)

```
<xs:complexType name="AddressesType">
  <xs:sequence>
    <xs:element name="Address" type="AddressType" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>Contain one client address</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

ELEMENT ADRESSESTYPE/ADDRESS

diagram



type
properties

AddressType

isRef 0
minOcc 1
maxOcc unbounded
content complex

children
annotation

[AddressLine1](#) [AddressLine2](#) [AddressLine3](#) [AddressLine4](#) [AddressLine5](#)

documentation

Contain one client address

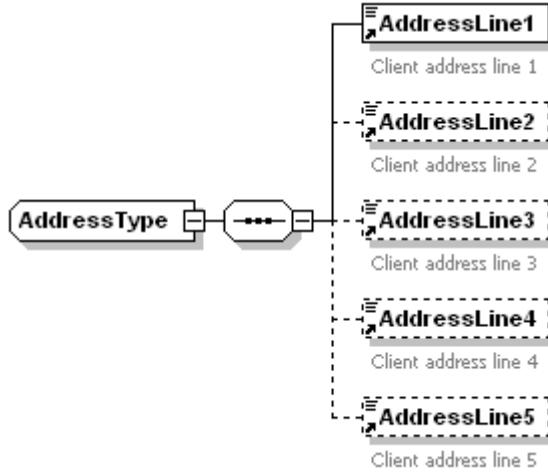
source

```
<xs:element name="Address" type="AddressType" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>Contain one client address</xs:documentation>
  </xs:annotation>
```

```
</xs:annotation>
</xs:element>
```

COMPLEXTYPE ADRESSTYPE

diagram



children
used by
source

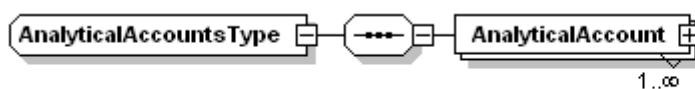
[AddressLine1](#) [AddressLine2](#) [AddressLine3](#) [AddressLine4](#) [AddressLine5](#)

```

element AddressType/Address
<xs:complexType name="AddressType">
  <xs:sequence>
    <xs:element ref="AddressLine1">
      <xs:annotation>
        <xs:documentation>Client address line 1</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="AddressLine2" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Client address line 2</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="AddressLine3" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Client address line 3</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="AddressLine4" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Client address line 4</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="AddressLine5" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Client address line 5</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

COMPLEXTYPE ANALYTICALACCOUNTSTYPE

diagram



Contain definition of one analytical account.

children

[AnalyticalAccount](#)

used by
annotation

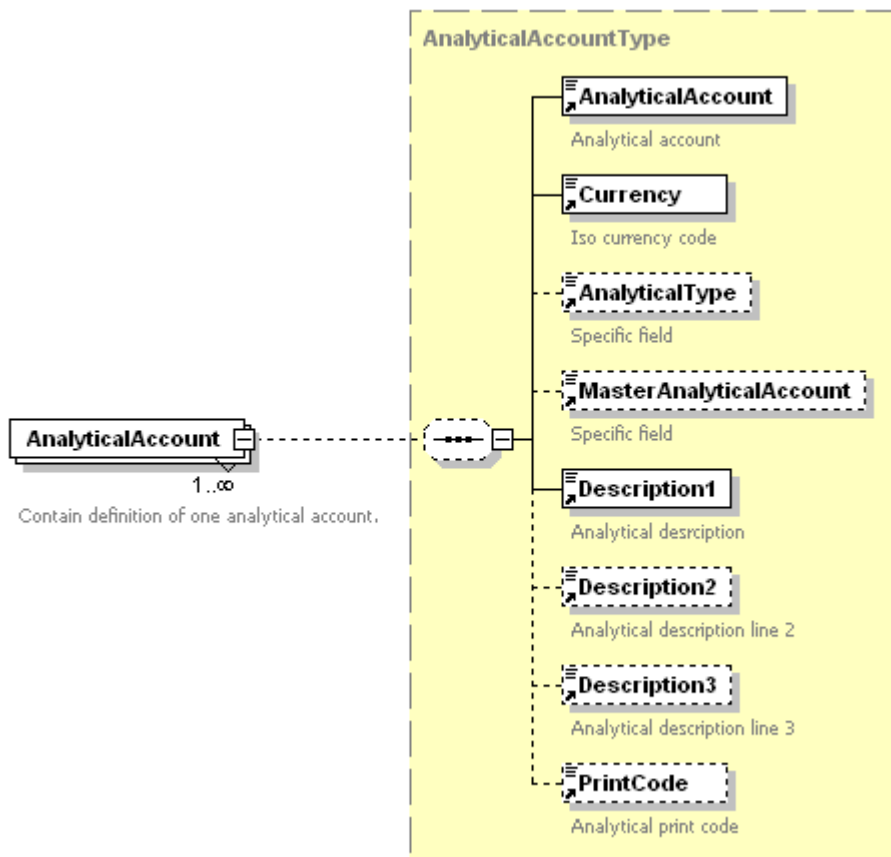
element [Xml2WinEUR/AnalyticalAccounts](#)
documentation

source

```
<xs:complexType name="AnalyticalAccountsType">
  <xs:annotation>
    <xs:documentation></xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="AnalyticalAccount" type="AnalyticalAccountType" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>Contain definition of one analytical account.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

ELEMENT ANALYTICALACCOUNTSTYPE/ANALYTICALACCOUNT

diagram



type
properties

[AnalyticalAccountType](#)

isRef 0
minOcc 1
maxOcc unbounded
content complex

children

[AnalyticalAccount](#) [Currency](#) [AnalyticalType](#) [MasterAnalyticalAccount](#) [Description1](#) [Description2](#) [Description3](#) [PrintCode](#)

used by
annotation

complexTypes [AnalyticalAccountType](#) [ClientLineType](#) [GeneralLedgerLineType](#) [InvoiceType](#)

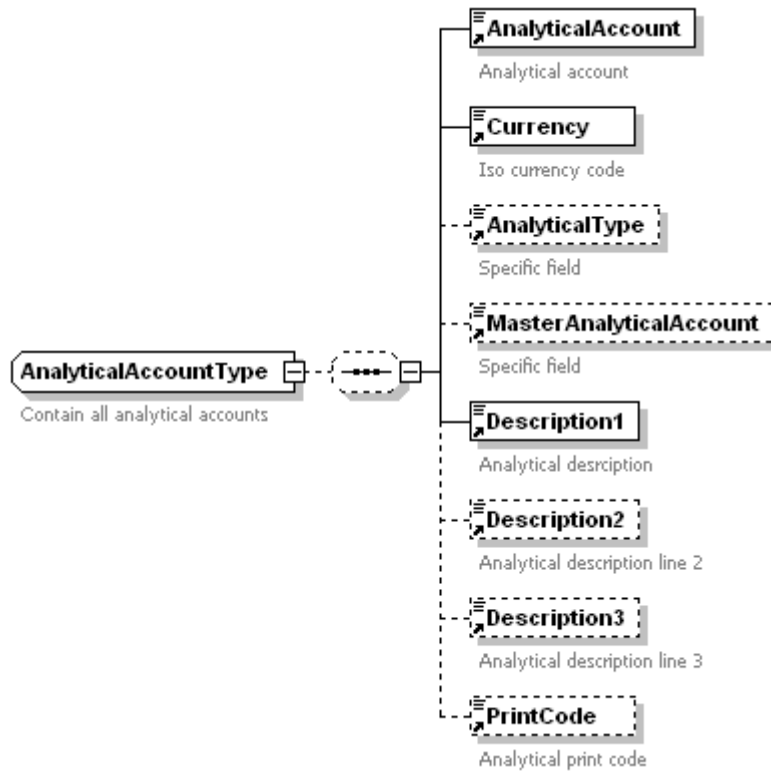
documentation
Contain definition of one analytical account.

source

```
<xs:element name="AnalyticalAccount" type="AnalyticalAccountType" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>Contain definition of one analytical account.</xs:documentation>
  </xs:annotation>
</xs:element>
```

COMPLEXTYPE ANALYTICALACCOUNTTYPE

diagram



children

[AnalyticalAccount](#) [Currency](#) [AnalyticalType](#) [MasterAnalyticalAccount](#) [Description1](#) [Description2](#) [Description3](#) [PrintCode](#)

used by
annotation

element [AnalyticalAccountsType/AnalyticalAccount](#)
documentation

source

```

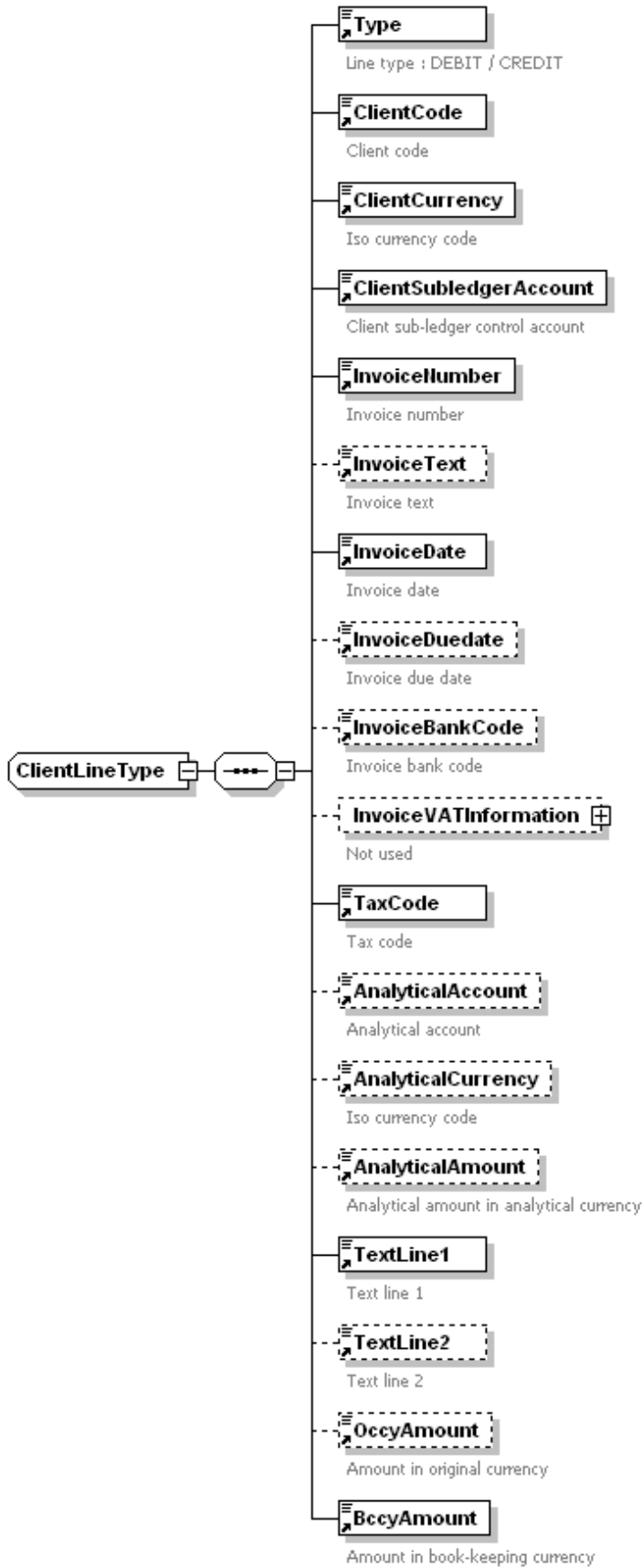
Contain all analytical accounts
<xs:complexType name="AnalyticalAccountType">
  <xs:annotation>
    <xs:documentation>Contain all analytical accounts</xs:documentation>
  </xs:annotation>
  <xs:sequence minOccurs="0">
    <xs:element ref="AnalyticalAccount"/>
    <xs:element ref="Currency"/>
    <xs:element ref="AnalyticalType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Specific field</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="MasterAnalyticalAccount" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Specific field</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="Description1">
      <xs:annotation>
        <xs:documentation>Analytical description</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="Description2" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Analytical description line 2</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="Description3" minOccurs="0">

```

```
<xs:annotation>  
  <xs:documentation>Analytical description line 3</xs:documentation>  
</xs:annotation>  
</xs:element>  
<xs:element ref="PrintCode" minOccurs="0">  
  <xs:annotation>  
    <xs:documentation>Analytical print code</xs:documentation>  
  </xs:annotation>  
</xs:element>  
</xs:sequence>  
</xs:complexType>
```

COMPLEXTYPE CLIENTLINETYPE

diagram



children

[Type](#) [ClientCode](#) [ClientCurrency](#) [ClientSubledgerAccount](#) [InvoiceNumber](#) [InvoiceText](#) [InvoiceDate](#) [InvoiceDuedate](#) [InvoiceBankCode](#) [InvoiceVATInformation](#) [TaxCode](#) [AnalyticalAccount](#) [AnalyticalCurrency](#) [AnalyticalAmount](#) [TextLine1](#) [TextLine2](#) [OccyAmount](#) [BccyAmount](#)

```

element EntryType/ClientLine
<xs:complexType name="ClientLineType">
  <xs:sequence>
    <xs:element ref="Type">
      <xs:annotation>
        <xs:documentation>Line type : DEBIT / CREDIT</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="ClientCode">
      <xs:annotation>
        <xs:documentation>Client code</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="ClientCurrency"/>
    <xs:element ref="ClientSubledgerAccount">
      <xs:annotation>
        <xs:documentation>Client sub-ledger control account</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="InvoiceNumber">
      <xs:annotation>
        <xs:documentation>Invoice number</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="InvoiceText" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Invoice text</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="InvoiceDate">
      <xs:annotation>
        <xs:documentation>Invoice date</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="InvoiceDuedate" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Invoice due date</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="InvoiceBankCode" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Invoice bank code</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="InvoiceVATInformation" type="InvoiceVATInformationType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Not used</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="TaxCode">
      <xs:annotation>
        <xs:documentation>Tax code</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="AnalyticalAccount" minOccurs="0"/>
    <xs:element ref="AnalyticalCurrency" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Iso currency code</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="AnalyticalAmount" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Analytical amount in analytical currency</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="TextLine1">
      <xs:annotation>
        <xs:documentation>Text line 1</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

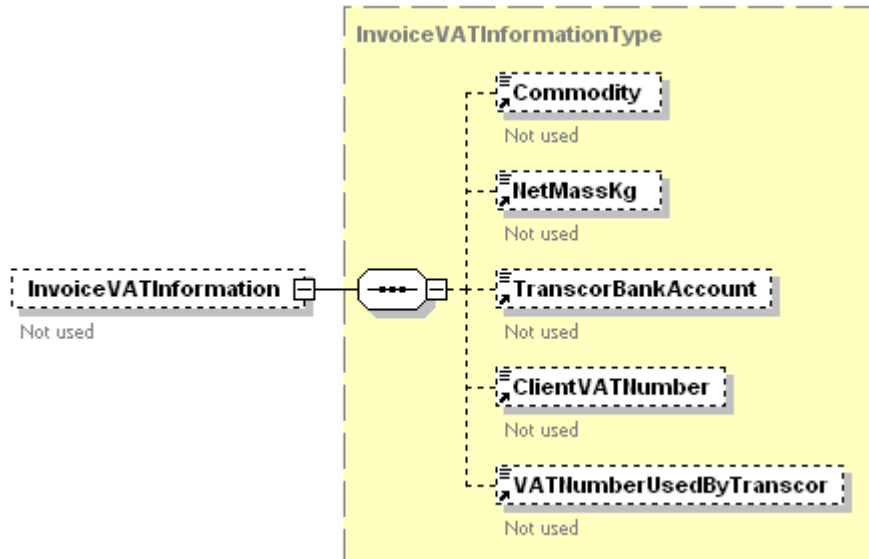
```

</xs:element>
<xs:element ref="TextLine2" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Text line 2</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="OccyAmount" minOccurs="0"/>
<xs:element ref="BccyAmount"/>
</xs:sequence>
</xs:complexType>

```

ELEMENT CLIENTLINETYPE/INVOICEVATINFORMATION

diagram



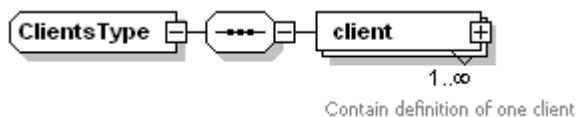
```

type
  InvoiceVATInformationType
properties
  isRef 0
  minOccurs 0
  maxOccurs 1
  content complex
children
  Commodity NetMassKg TranscorBankAccount ClientVATNumber VATNumberUsedByTranscor
annotation
  documentation
  Not used
source
<xs:element name="InvoiceVATInformation" type="InvoiceVATInformationType" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Not used</xs:documentation>
  </xs:annotation>
</xs:element>

```

COMPLEXTYPE CLIENTSTYPE

diagram



```

children
  used by
  source
  client
  element Xml2WinEUR/Clients
<xs:complexType name="ClientsType">
  <xs:sequence>
    <xs:element name="client" type="ClientType" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>Contain definition of one client</xs:documentation>

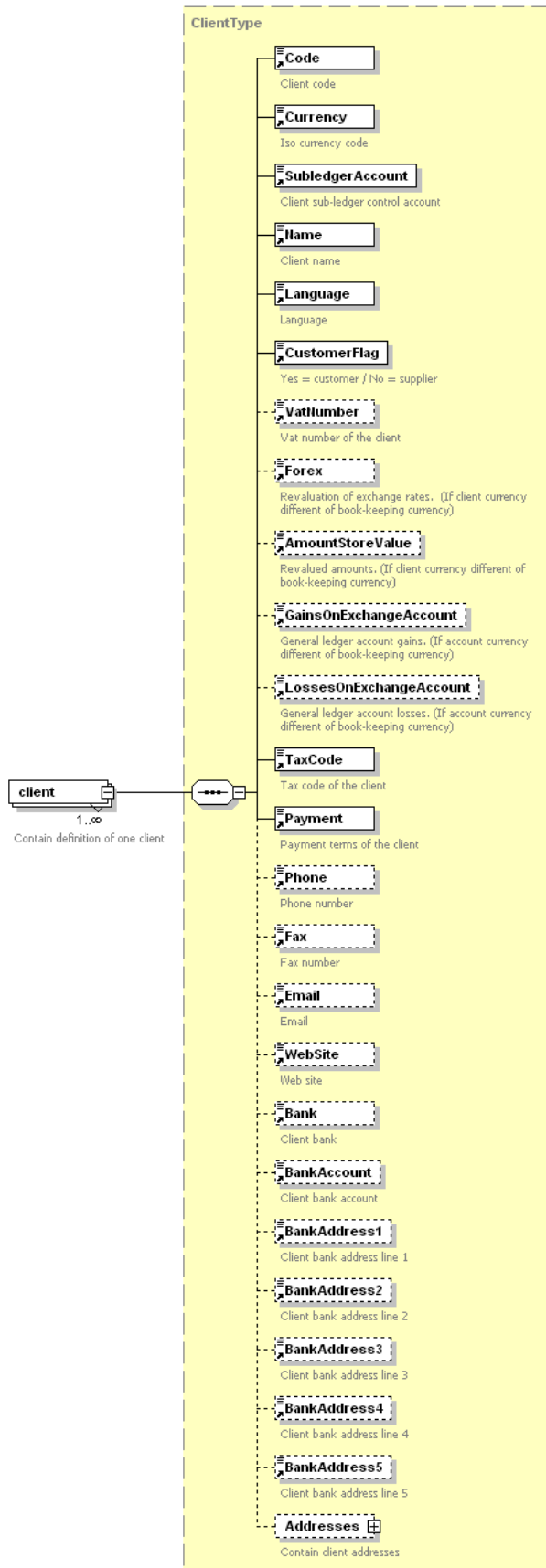
```




```
</xs:annotation>  
</xs:element>  
</xs:sequence>  
</xs:complexType>
```

ELEMENT CLIENTSTYPE/CLIENT

diagram



type	ClientType	
properties	isRef	0
	minOcc	1



maxOcc unbounded
content complex

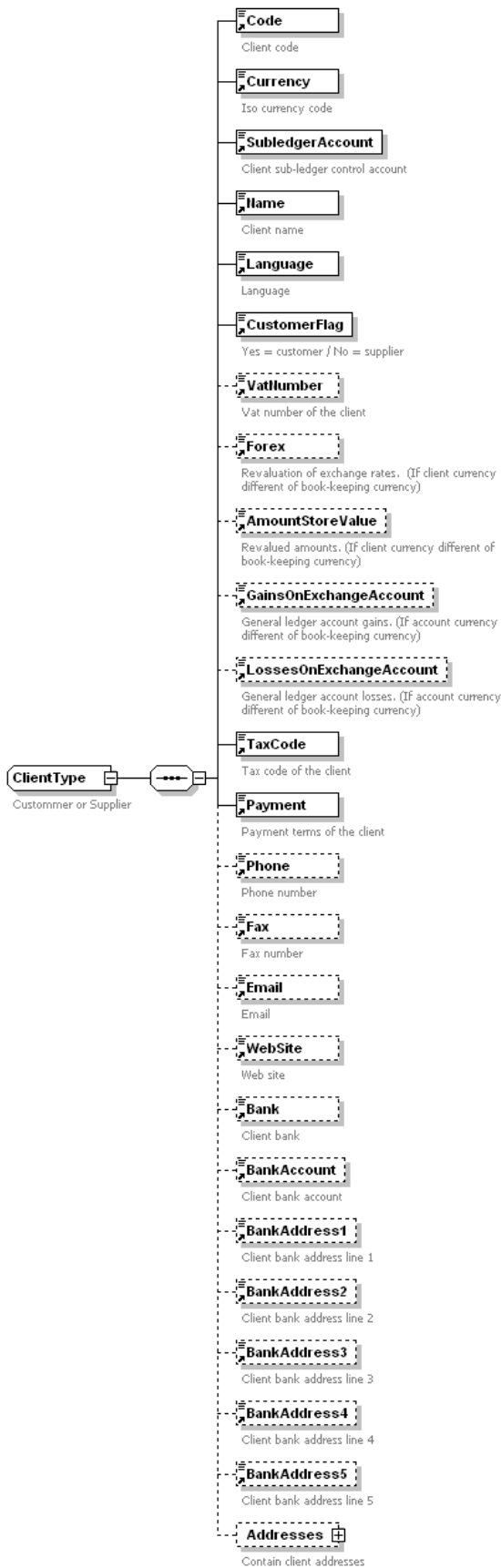
children [Code](#) [Currency](#) [SubledgerAccount](#) [Name](#) [Language](#) [CustomerFlag](#) [VatNumber](#) [Forex](#) [AmountStoreValue](#) [GainsOnExchangeAccount](#) [LossesOnExchangeAccount](#) [TaxCode](#) [Payment](#) [Phone](#) [Fax](#) [Email](#) [WebSite](#) [Bank](#) [BankAccount](#) [BankAddress1](#) [BankAddress2](#) [BankAddress3](#) [BankAddress4](#) [BankAddress5](#) [Addresses](#)

annotation documentation
Contain definition of one client

source `<xs:element name="client" type="ClientType" maxOccurs="unbounded">
<xs:annotation>
 <xs:documentation>Contain definition of one client</xs:documentation>
</xs:annotation>
</xs:element>`

COMPLEXTYPE CLIENTTYPE

diagram



children

[Code](#) [Currency](#) [SubledgerAccount](#) [Name](#) [Language](#) [CustomerFlag](#) [VatNumber](#) [Forex](#) [AmountStoreValue](#) [GainsOnExchangeAccount](#) [LossesOnExchangeAccount](#) [TaxCode](#) [Payment](#) [Phone](#) [Fax](#) [Email](#) [WebSite](#) [Bank](#) [BankAccount](#) [BankAddress1](#) [BankAddress2](#) [BankAddress3](#) [BankAddress4](#) [BankAddress5](#) [Addresses](#)

used by
annotation

source

element [ClientsType/client](#)

documentation

Customer or Supplier

```

<xs:complexType name="ClientType">
  <xs:annotation>
    <xs:documentation>Customer or Supplier</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element ref="Code">
      <xs:annotation>
        <xs:documentation>Client code</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="Currency"/>
    <xs:element ref="SubledgerAccount">
      <xs:annotation>
        <xs:documentation>Client sub-ledger control account</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="Name">
      <xs:annotation>
        <xs:documentation>Client name</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="Language">
      <xs:annotation>
        <xs:documentation>Language</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="CustomerFlag">
      <xs:annotation>
        <xs:documentation>Yes = customer / No = supplier</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="VatNumber" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Vat number of the client</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="Forex" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Revaluation of exchange rates. (If client currency different of book-keeping currency)</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="AmountStoreValue" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Revalued amounts. (If client currency different of book-keeping currency)</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="GainsOnExchangeAccount" minOccurs="0">
      <xs:annotation>
        <xs:documentation>General ledger account gains. (If account currency different of book-keeping currency)</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="LossesOnExchangeAccount" minOccurs="0">
      <xs:annotation>
        <xs:documentation>General ledger account losses. (If account currency different of book-keeping currency)</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="TaxCode">
      <xs:annotation>
        <xs:documentation>Tax code of the client</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="Payment">
      <xs:annotation>
        <xs:documentation>Payment terms of the client</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

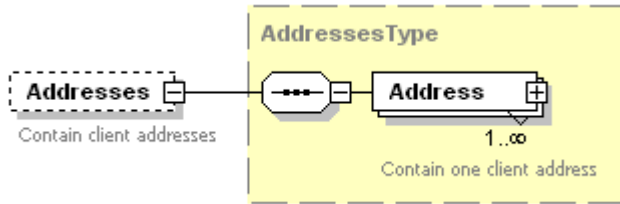
```

```

<xs:element ref="Phone" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Phone number</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="Fax" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Fax number</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="Email" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Email</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="WebSite" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Web site</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="Bank" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Client bank</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="BankAccount" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Client bank account</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="BankAddress1" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Client bank address line 1</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="BankAddress2" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Client bank address line 2</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="BankAddress3" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Client bank address line 3</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="BankAddress4" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Client bank address line 4</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="BankAddress5" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Client bank address line 5</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Addresses" type="AddressesType" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Contain client addresses</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>

```

diagram



type
properties

AddressesType
isRef 0
minOcc 0
maxOcc 1
content complex

children
annotation

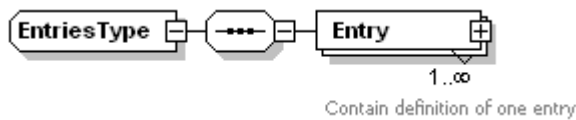
Address
documentation
Contain client addresses

source

```
<xs:element name="Addresses" type="AddressesType" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Contain client addresses</xs:documentation>
  </xs:annotation>
</xs:element>
```

COMPLEXTYPE ENTRIESTYPE

diagram

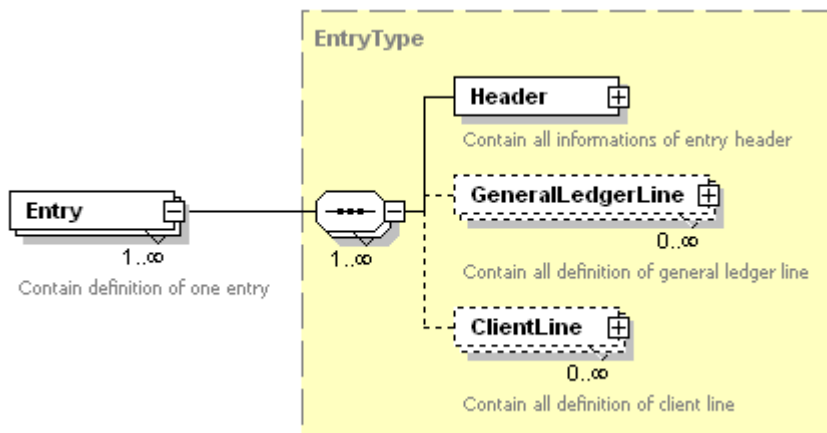


children
used by
source

Entry
element [Xml2WinEUR/Entries](#)
<xs:complexType name="EntriesType">
 <xs:sequence>
 <xs:element name="Entry" type="EntryType" maxOccurs="unbounded">
 <xs:annotation>
 <xs:documentation>Contain definition of one entry</xs:documentation>
 </xs:annotation>
 </xs:element>
 </xs:sequence>
</xs:complexType>

ELEMENT ENTRIESTYPE/ENTRY

diagram



type
properties

EntryType
isRef 0
minOcc 1

children annotation source

maxOcc content

unbounded complex

[Header](#) [GeneralLedgerLine](#) [ClientLine](#)

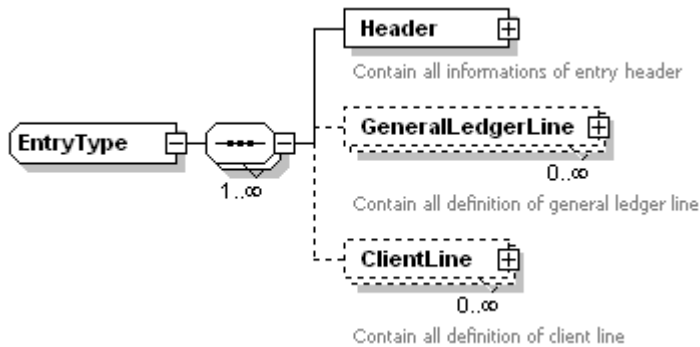
documentation

Contain definition of one entry

```
<xs:element name="Entry" type="EntryType" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>Contain definition of one entry</xs:documentation>
  </xs:annotation>
</xs:element>
```

COMPLEXTYPE ENTRYTYPE

diagram



children used by source

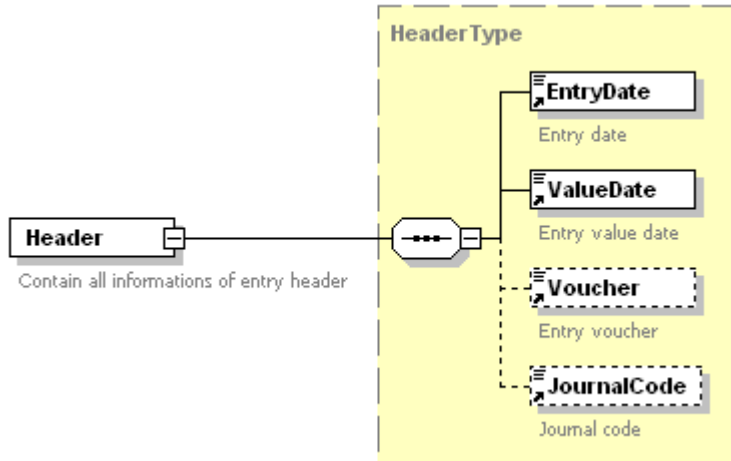
[Header](#) [GeneralLedgerLine](#) [ClientLine](#)

element **[EntryType/Entry](#)**

```
<xs:complexType name="EntryType">
  <xs:sequence maxOccurs="unbounded">
    <xs:element name="Header" type="HeaderType">
      <xs:annotation>
        <xs:documentation>Contain all informations of entry header</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="GeneralLedgerLine" type="GeneralLedgerLineType" minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>Contain all definition of general ledger line</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="ClientLine" type="ClientLineType" minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>Contain all definition of client line</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

ELEMENT ENTRYTYPE/HEADER

diagram



type
properties

HeaderType

isRef 0
content complex

children
annotation

[EntryDate](#) [ValueDate](#) [Voucher](#) [JournalCode](#)

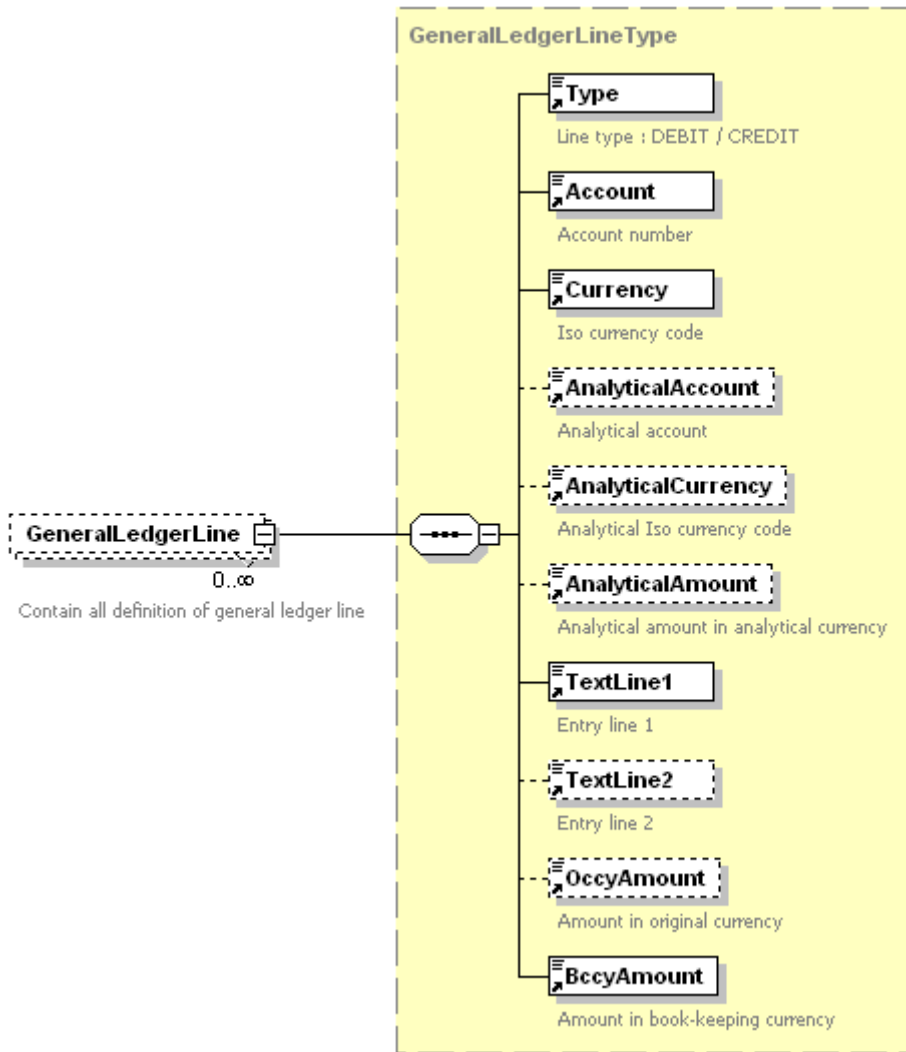
documentation
Contain all informations of entry header

source

```

<xs:element name="Header" type="HeaderType">
  <xs:annotation>
    <xs:documentation>Contain all informations of entry header</xs:documentation>
  </xs:annotation>
</xs:element>
  
```

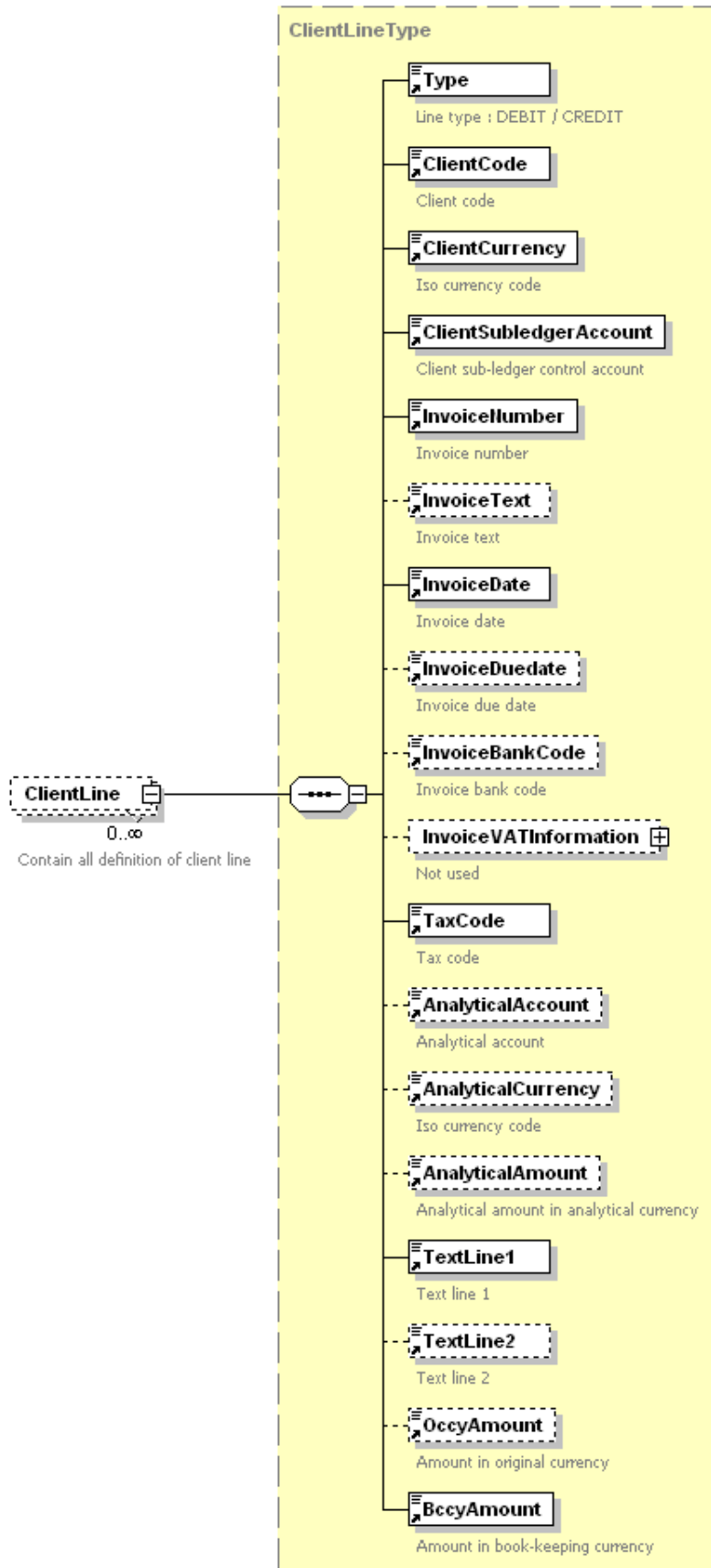
ELEMENT ENTRYTYPE/GENERALLEDGERLINE



type	GeneralLedgerLineType
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	Type Account Currency AnalyticalAccount AnalyticalCurrency AnalyticalAmount TextLine1 TextLine2 OccyAmount BccyAmount
annotation	documentation
source	Contain all definition of general ledger line <pre> <xs:element name="GeneralLedgerLine" type="GeneralLedgerLineType" minOccurs="0" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>Contain all definition of general ledger line</xs:documentation> </xs:annotation> </xs:element> </pre>

ELEMENT ENTRYTYPE/CLIENTLINE

diagram

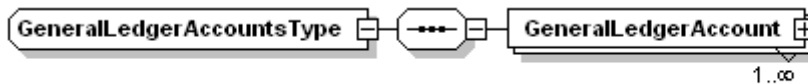


type **ClientLineType**
 properties
 isRef 0
 minOcc 0

	maxOcc	unbounded
	content	complex
children	Type ClientCode ClientCurrency ClientSubledgerAccount InvoiceNumber InvoiceText InvoiceDate InvoiceDuedate InvoiceBankCode InvoiceVATInformation TaxCode AnalyticalAccount AnalyticalCurrency AnalyticalAmount TextLine1 TextLine2 OccyAmount BccyAmount	
annotation	documentation	
	Contain all definition of client line	
source	<pre><xs:element name="ClientLine" type="ClientLineType" minOccurs="0" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>Contain all definition of client line</xs:documentation> </xs:annotation> </xs:element></pre>	

COMPLEXTYPE GENERALLEDGERACCOUNTSTYPE

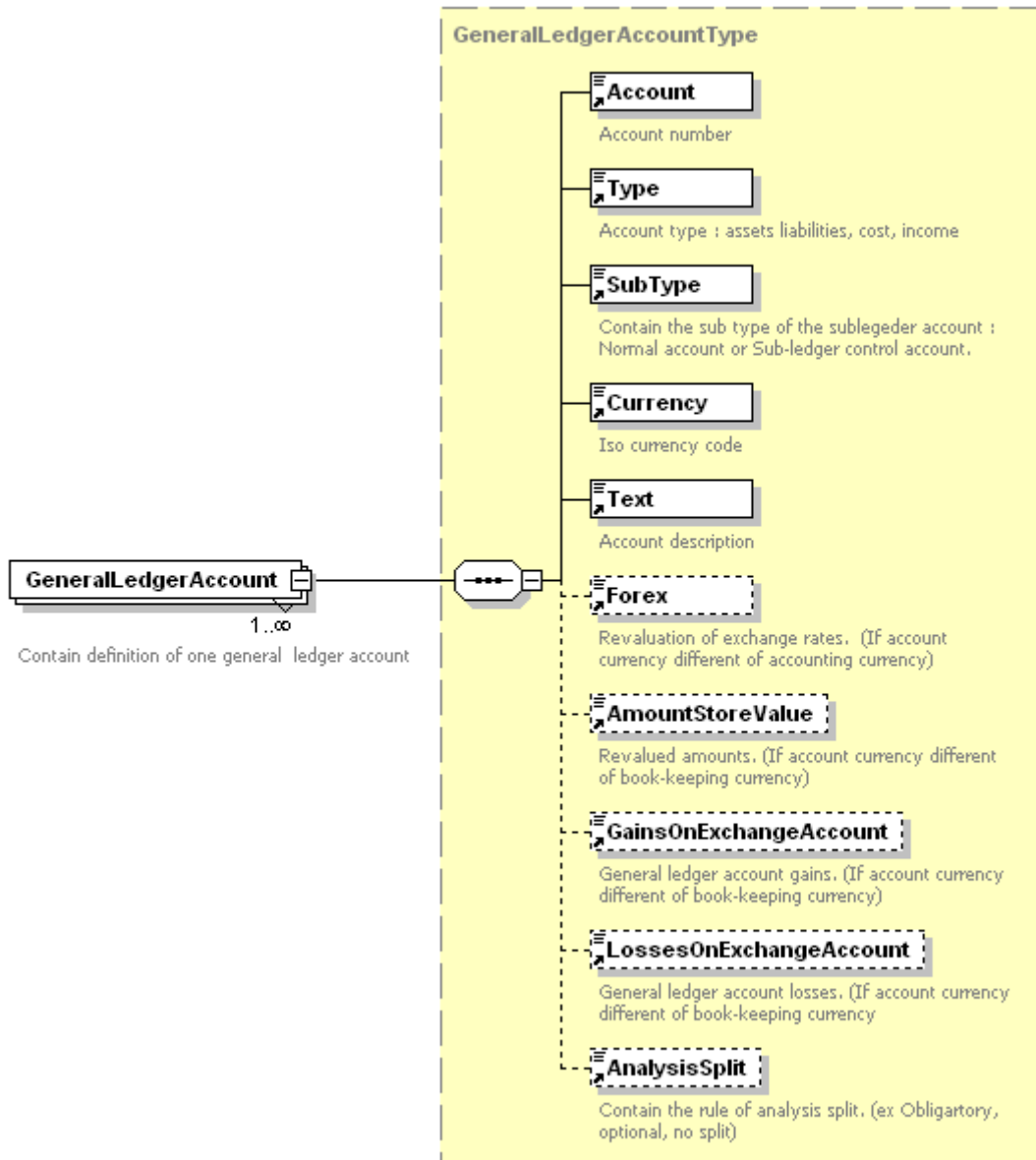
diagram



Contain definition of one general ledger account

children	GeneralLedgerAccount	
used by	element Xml2WinEUR/GeneralLedgerAccounts	
source	<pre><xs:complexType name="GeneralLedgerAccountsType"> <xs:sequence> <xs:element name="GeneralLedgerAccount" type="GeneralLedgerAccountType" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>Contain definition of one general ledger account</xs:documentation> </xs:annotation> </xs:element> </xs:sequence> </xs:complexType></pre>	

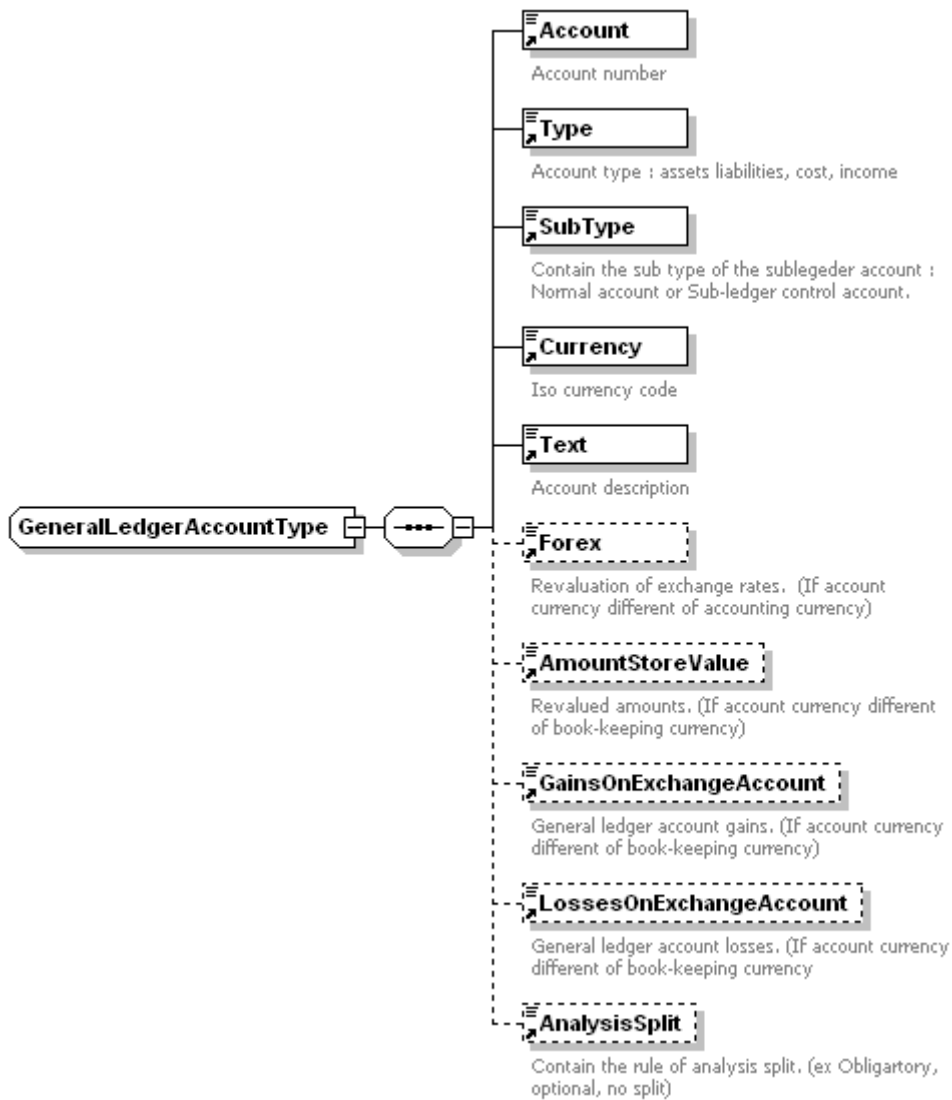
ELEMENT GENERALLEDGERACCOUNTSTYPE/GENERALLEDGERACCOUNT



type	GeneralLedgerAccountType								
properties	<table border="0"> <tr><td>isRef</td><td>0</td></tr> <tr><td>minOcc</td><td>1</td></tr> <tr><td>maxOcc</td><td>unbounded</td></tr> <tr><td>content</td><td>complex</td></tr> </table>	isRef	0	minOcc	1	maxOcc	unbounded	content	complex
isRef	0								
minOcc	1								
maxOcc	unbounded								
content	complex								
children	Account Type SubType Currency Text Forex AmountStoreValue GainsOnExchangeAccount LossesOnExchangeAccount AnalysisSplit								
annotation	documentation								
source	<pre> <xs:element name="GeneralLedgerAccount" type="GeneralLedgerAccountType" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>Contain definition of one general ledger account</xs:documentation> </xs:annotation> </xs:element> </pre>								

COMPLEXTYPE GENERALLEDGERACCOMNTTYPE

diagram



children

[Account](#) [Type](#) [SubType](#) [Currency](#) [Text](#) [Forex](#) [AmountStoreValue](#) [GainsOnExchangeAccount](#) [LossesOnExchangeAccount](#) [AnalysisSplit](#)

used by
source

element [GeneralLedgerAccountsType/GeneralLedgerAccount](#)

```

<xs:complexType name="GeneralLedgerAccountType">
  <xs:sequence>
    <xs:element ref="Account"/>
    <xs:element ref="Type">
      <xs:annotation>
        <xs:documentation>Account type : assets liabilities, cost, income</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="SubType">
      <xs:annotation>
        <xs:documentation>Contain the sub type of the subledger account : Normal account or Sub-ledger control
account.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="Currency"/>
    <xs:element ref="Text">
      <xs:annotation>
        <xs:documentation>Account description</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="Forex" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Revaluation of exchange rates. (If account currency different of accounting currency)</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

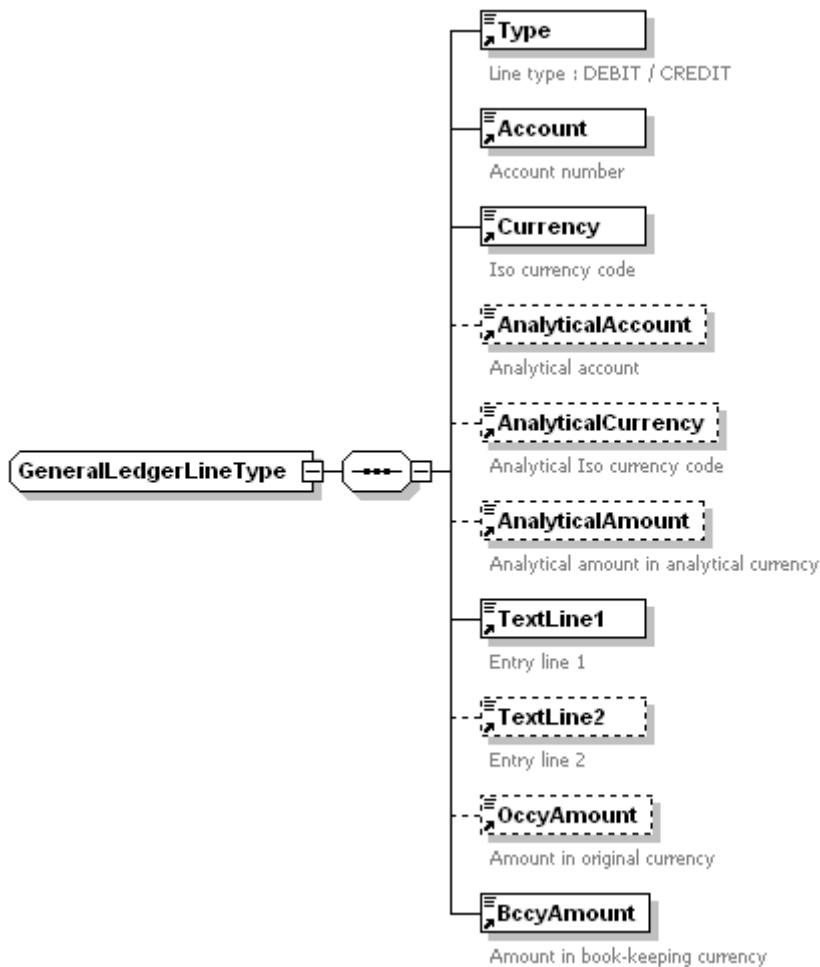
```

</xs:element>
<xs:element ref="AmountStoreValue" minOccurs="0">
<xs:annotation>
  <xs:documentation>Revalued amounts. (If account currency different of book-keeping currency)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element ref="GainsOnExchangeAccount" minOccurs="0">
<xs:annotation>
  <xs:documentation>General ledger account gains. (If account currency different of book-keeping currency)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element ref="LossesOnExchangeAccount" minOccurs="0">
<xs:annotation>
  <xs:documentation>General ledger account losses. (If account currency different of book-keeping currency)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element ref="AnalysisSplit" minOccurs="0">
<xs:annotation>
  <xs:documentation>Contain the rule of analysis split. (ex Obligatory, optional, no split)</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>

```

COMPLEXTYPE GENERALLEDGERLINETYPE

diagram



children
used by

[Type](#) [Account](#) [Currency](#) [AnalyticalAccount](#) [AnalyticalCurrency](#) [AnalyticalAmount](#) [TextLine1](#) [TextLine2](#) [OccyAmount](#) [BccyAmount](#)
 element [EntryType/GeneralLedgerLine](#)

source

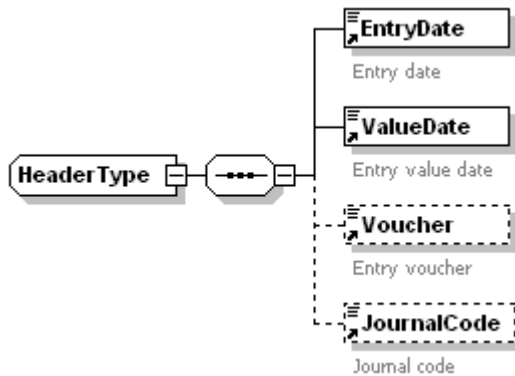
```

<xs:complexType name="GeneralLedgerLineType">
  <xs:sequence>
    <xs:element ref="Type">
      <xs:annotation>
        <xs:documentation>Line type : DEBIT / CREDIT</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="Account"/>
    <xs:element ref="Currency"/>
    <xs:element ref="AnalyticalAccount" minOccurs="0"/>
    <xs:element ref="AnalyticalCurrency" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Analytical Iso currency code</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="AnalyticalAmount" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Analytical amount in analytical currency</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="TextLine1">
      <xs:annotation>
        <xs:documentation>Entry line 1</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="TextLine2" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Entry line 2</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="OccyAmount" minOccurs="0"/>
    <xs:element ref="BccyAmount"/>
  </xs:sequence>
</xs:complexType>

```

COMPLEXTYPE HEADERTYPE

diagram



children
used by
source

[EntryDate](#) [ValueDate](#) [Voucher](#) [JournalCode](#)

element

[EntryType/Header](#)

```

<xs:complexType name="HeaderType">
  <xs:sequence>
    <xs:element ref="EntryDate">
      <xs:annotation>
        <xs:documentation>Entry date</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="ValueDate">
      <xs:annotation>
        <xs:documentation>Entry value date</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>

```



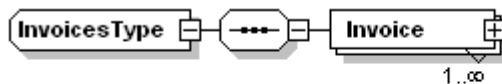
```

<xs:element ref="Voucher" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Entry voucher</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="JournalCode" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Journal code</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>

```

COMPLEXTYPE INVOICESTYPE

diagram



Not used for import. Contain one definition of invoice

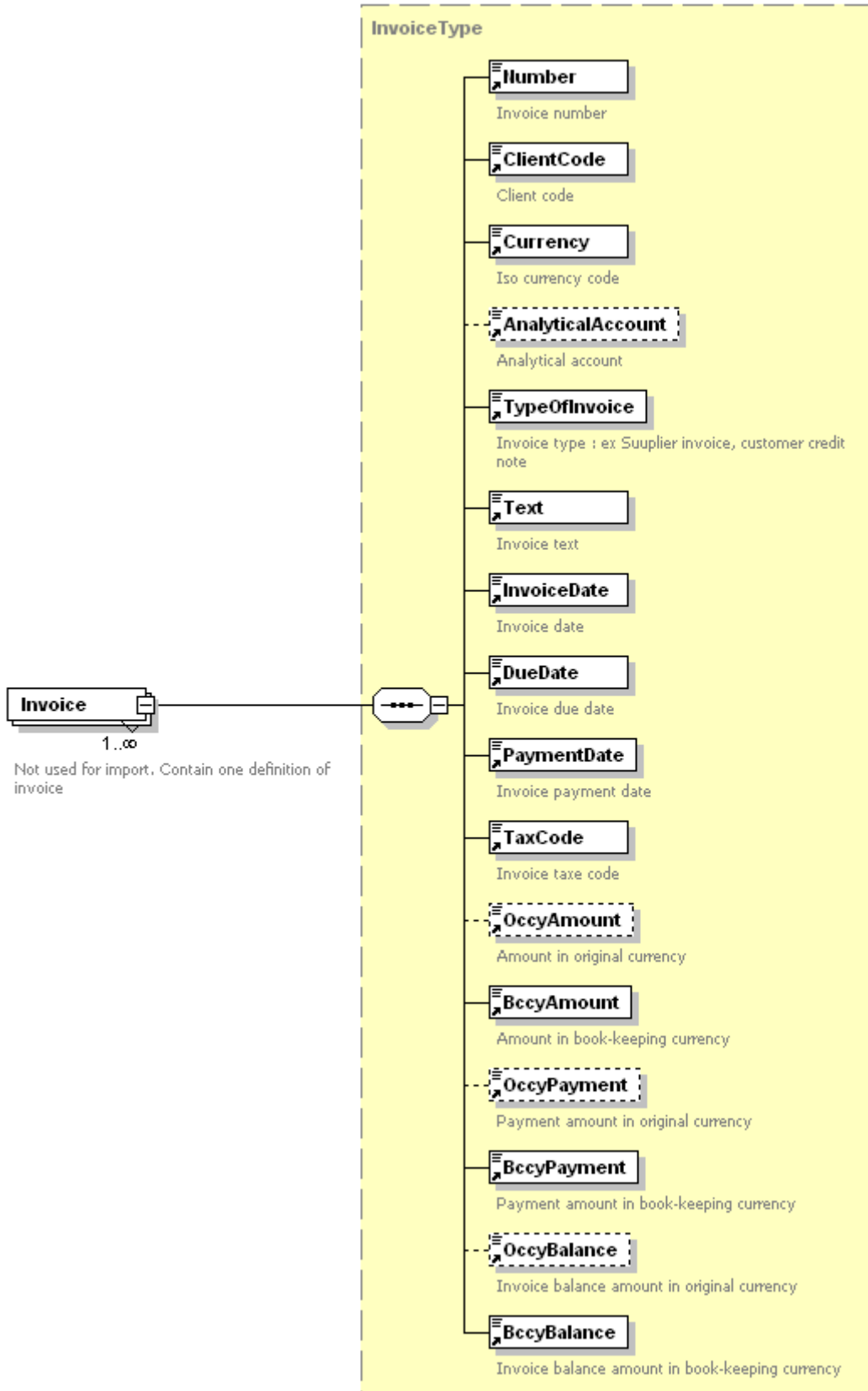
children
used by
source

```

Invoice
element Xml2WinEUR/Invoices
<xs:complexType name="InvoicesType">
  <xs:sequence>
    <xs:element name="Invoice" type="InvoiceType" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>Not used for import. Contain one definition of invoice</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

ELEMENT INVOICESTYPE/INVOICE



type	InvoiceType	
properties	isRef	0
	minOcc	1
	maxOcc	unbounded
	content	complex



children

[Number](#) [ClientCode](#) [Currency](#) [AnalyticalAccount](#) [TypeOfInvoice](#) [Text](#) [InvoiceDate](#) [DueDate](#) [PaymentDate](#) [TaxCode](#)
[OccyAmount](#) [BccyAmount](#) [OccyPayment](#) [BccyPayment](#) [OccyBalance](#) [BccyBalance](#)

annotation

documentation

Not used for import. Contain one definition of invoice

source

```
<xs:element name="Invoice" type="InvoiceType" maxOccurs="unbounded">
```

```
<xs:annotation>
```

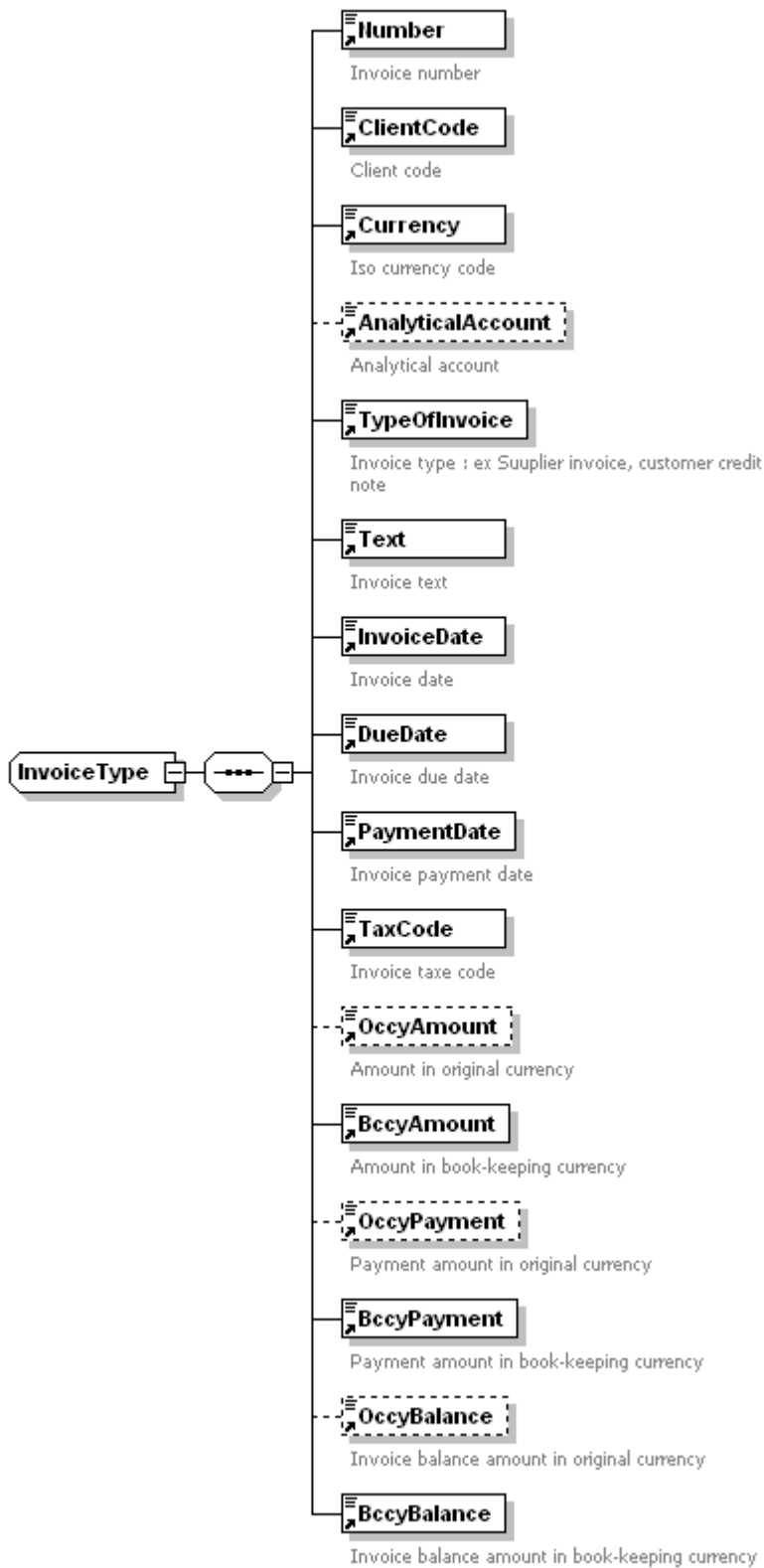
```
<xs:documentation>Not used for import. Contain one definition of invoice</xs:documentation>
```

```
</xs:annotation>
```

```
</xs:element>
```

COMPLEXTYPE INVOICETYPE

diagram



children

used by
source

[Number](#) [ClientCode](#) [Currency](#) [AnalyticalAccount](#) [TypeOfInvoice](#) [Text](#) [InvoiceDate](#) [DueDate](#) [PaymentDate](#) [TaxCode](#) [OccyAmount](#) [BccyAmount](#) [OccyPayment](#) [BccyPayment](#) [OccyBalance](#) [BccyBalance](#)

element [InvoicesType/Invoice](#)

```
<xs:complexType name="InvoiceType">
```

```
<xs:sequence>
```

```
<xs:element ref="Number">
```

```
<xs:annotation>
```

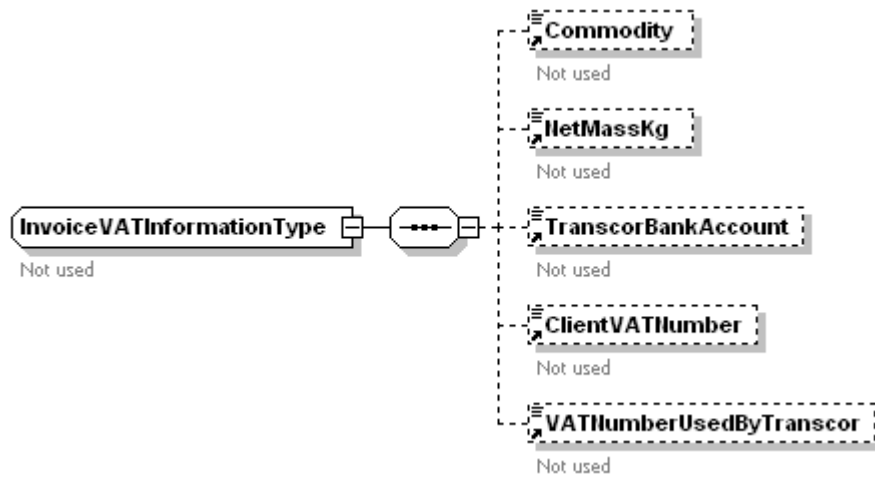
```
<xs:documentation>Invoice number</xs:documentation>
```

```

</xs:annotation>
</xs:element>
<xs:element ref="ClientCode">
  <xs:annotation>
    <xs:documentation>Client code</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="Currency"/>
<xs:element ref="AnalyticalAccount" minOccurs="0"/>
<xs:element ref="TypeOfInvoice">
  <xs:annotation>
    <xs:documentation>Invoice type : ex Suuplier invoice, customer credit note</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="Text">
  <xs:annotation>
    <xs:documentation>Invoice text</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="InvoiceDate">
  <xs:annotation>
    <xs:documentation>Invoice date</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="DueDate">
  <xs:annotation>
    <xs:documentation>Invoice due date</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="PaymentDate">
  <xs:annotation>
    <xs:documentation>Invoice payment date</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="TaxCode">
  <xs:annotation>
    <xs:documentation>Invoice tax code</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="OccyAmount" minOccurs="0"/>
<xs:element ref="BccyAmount"/>
<xs:element ref="OccyPayment" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Payment amount in original currency</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="BccyPayment">
  <xs:annotation>
    <xs:documentation>Payment amount in book-keeping currency</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="OccyBalance" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Invoice balance amount in original currency</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element ref="BccyBalance">
  <xs:annotation>
    <xs:documentation>Invoice balance amount in book-keeping currency</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>

```

diagram



children
used by
annotation

[Commodity](#) [NetMassKg](#) [TranscorBankAccount](#) [ClientVATNumber](#) [VATNumberUsedByTranscor](#)
 element [ClientLineType/InvoiceVATInformation](#)

source

```

<xs:complexType name="InvoiceVATInformationType">
  <xs:annotation>
    <xs:documentation>Not used</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element ref="Commodity" minOccurs="0"/>
    <xs:element ref="NetMassKg" minOccurs="0"/>
    <xs:element ref="TranscorBankAccount" minOccurs="0"/>
    <xs:element ref="ClientVATNumber" minOccurs="0"/>
    <xs:element ref="VATNumberUsedByTranscor" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
  
```

XML Schema documentation generated by [XMLSpy](http://www.altova.com/xmlspy) Schema Editor <http://www.altova.com/xmlspy>