

**Alphabet Rosetta accounting file format and method**

Alphabet Rosetta accounting file format and method ..... 1

1. Open document format and methods ..... 3

2. Scope ..... 3

3. Brief ..... 3

4. History Rosetta ..... 5

5. How did the SIE4 format become so successful? ..... 5

6. Accounting interchange methods – What is that? ..... 6

7. The market of the common man is pushing ..... 7

8. The SIE/Alphabet Rosetta accounting methods ..... 7

9. The SIE/Alphabet Rosetta accounting object methods ..... 8

10. The Rosetta accounting file format – specification ..... 13

10.1. The AccountingType ..... 13

The file header information ..... 15

10.2. The AttestType ..... 16

10.3. The PartyType ..... 17

10.4. The ContactType ..... 18

Accounts Chart ..... 19

10.5. The AccountType ..... 19

10.6. The AccountCodeType ..... 21

Object Chart ..... 22

10.7. The DimensionType ..... 22

10.8. Dimension ObjectType ..... 23

Ledgers Chart ..... 24

10.9. The LedgerType ..... 24

10.10. The BalanceReportType ..... 25

10.11. The BalanceTotalType ..... 26

10.12. The BalanceTotalPeriodType ..... 26

10.13. The BalanceAccountType ..... 27

10.14. The BalanceAccountPeriodType ..... 28

10.15. The VoucherType ..... 29

10.16. The VoucherEntryType ..... 30

11.	The Rosetta digital Invoice file format – specification .....	32
11.1.	The interpreting and process ability issue .....	32
11.2.	Theoretic basis of what is an Invoice .....	33
11.3.	Formalised procurement needs .....	33
11.4.	The need of being accurate in legal courts .....	33
11.5.	Customer needs of purchase information, use, repair, security and disposal information.....	35
11.6.	The information distribution need in Web-trade .....	35
11.7.	The support of the computer industry .....	36
11.8.	UN/CEFACT documents.....	36
11.9.	XBRL.....	38
11.10.	Invoice link - The Accounting Voucher or its entry .....	38
11.11.	EU VAT directive .....	39
11.12.	The basis for an accounting electronic invoice format.....	40
11.13.	The InvoiceType.....	42
11.14.	The InvoiceEntryType.....	45
11.15.	The TradeItemDataType.....	48
11.16.	The PayType.....	48
11.17.	The DepreciationsType.....	49
11.18.	The TextType.....	50
12.	The Rosetta accounting file I/O open library.....	50

## 1. Open document format and methods

The Alphabet Rosetta accounting format and method as well as the Alphabet Rosetta accounting XBRL/GL Taxonomy are free and open for anyone to use:

- <http://www.alphabet.se/Rosetta/Rosetta.xsd>
- [http://www.alphabet.se/Rosetta/Rosetta\\_Specification.pdf](http://www.alphabet.se/Rosetta/Rosetta_Specification.pdf)
- [http://www.alphabet.se/XBRL/Rosetta\\_XBRL-GL\\_Taxonomy.pdf](http://www.alphabet.se/XBRL/Rosetta_XBRL-GL_Taxonomy.pdf)

There are no Copyright or IPR demands on the file format, methods and use of these three documents and its content, from our side.

The only requirements are attribution, referring to Alphabet AB Rosetta when used.

## 2. Scope

This document describes the Alphabet Rosetta accounting file format and method.

The Alphabet Rosetta accounting file format and method is used as the main basis of the Alphabet Rosetta Accounting OS-independent C-Engine basic module. It is the basis of Alphabet Rosetta Accounting programs / apps in any OS and is for possible OEM sales.

The specification of interactivity with XBRL/GL is specified in the document “Alphabet Rosetta accounting XBRL-GL Taxonomy”.

The Alphabet Rosetta accounting file format and method is based on the methods of the SIE4 specification (retrievable in English from <http://www.sie.se/>). The Rosetta format is an extended SIE4 in XML.

The Alphabet Rosetta accounting file format XML schema is located at: <http://www.alphabet.se/Rosetta/Rosetta.xsd>

The [Rosetta.xsd](#) includes report title texts in the supported languages in order to allow XML script users to be able to create complete reports by just making the layouts and retrieve the text and report data.

Samples in this document origin from the SIE official sample SIE4.se.

Special thanks in understanding computer theory of accounting and computer formats (even though we all do not agree of all) to, the later SIE year meeting and its future file format discussions, Margareta Gandy Dagens industri, Robert Lemense Edificas, of the UN/CEFACT-TBG12 and Gianluca Garbellotto XBRL/GL International.

## 3. Brief

The Alphabet Rosetta accounting file format and method and the basic module is designed to be able to:

- Continue the successful use of SIE and develop the basic idea and concept. Be logically compatible with SIE4 (not SIE5 that is non-compatible with the successful SIE4 and not legal for international use),
  - Main purpose is entering the complete digital documentation world
- Presenting a easy to understand and easy readable basic XML-data files of accounting, audit and business administration
- Use the experience and the thoughts from XBRL/GL and UN/CEFACT business administrative standard digital document projects. Be logically compatible with XBRL/GL and

- adaptive in use of UN/CEFACT digital documents outside the UN/CEFACT accounting specifications (that is another scope).
- Harmonise with the EU VAT directive 2006/112/EG, especially the specification and logical concept of:
    - The Invoice (article 230)
    - The Receipt (article 226b)
    - The Digital invoice (article 233)
    - The conceptual idea and logics that the following also are invoice implementations in documenting:
      - Clearing and Bank account statements
      - Tax reports (self-billed)
      - Salary statements (self-billed)
      - Receivables (a Clearing account)
      - Payables (a Clearing account)
      - Pay order (a Clearing account)
      - Procurement documentation
    - The VAT directive is such a good basic documentation that it is expected to be copied also in non-EU member states. Copying/relating to it has major advantages.
  - The general needs of the commercial trade, especially web-based atomised trade systems
    - Product type information
    - Product serial number information
    - Distribution of product information/digital product sheets (for web-trade sites)
    - Support for factoring and Invoice collection
    - Support of procurement procedures
    - Making supplies transparently traceable
    - Supporting automated supply chains (ability to design automated web trade administrative systems)
      - And as such supporting semi-automated versions like a physical shop administration
    - Safety and environmental handling information to be distributed in the supply chain handling
    - Transparency to the nuts and bolts with logistics and integration of logistics and accounting inventories.
  - The general need of customers getting purchase information retrievable from accounting and business documentation. Supply of digital invoices and digital receipts with high information standards. Being able to retrieve:
    - Product type information
    - Product serial number information
    - Digital product sheets
    - Guarantee sheets
    - User registration information
    - Manuals and user instruction sheet
    - Technical support documents
    - Faulty product handling
    - Repairs and maintenance information
    - Safety and environmental handling information in use of the products
    - Recycling information
  - Multiple language support:
    - For international trade (Multilingual invoices)
    - For countries with multiple language markets (multilingual invoice texts and product information)
    - For international business interaction (being able to read accounting over the borders)

## 4. History Rosetta

Jan Bergström the initial owner of Alphabet AB was 2006-2008 part time remunerated secretary of the SIE accounting file format information exchange vendor/user interest group. As such substitute in the board of XBRL Sweden 2007, representing SIE at the BAS yearly meeting and in general organising the work of the SIE board.

The period was after 10 years of relative calmness, after the internationally unparalleled success of the SIE4 format developed in the first half of the 1990-ies. The board decided to check out the ambitions, possibilities and opportunities to form a plan for the future work.

This included in taking contact with governmental development projects partly on initiative from the EU commission and other processes domestically. This including the advantages and disadvantages of becoming a formalised standard organisation. As a result SIE got representation of the Swedish GAAP authority BFN.

We had contacts with digital invoicing projects and especially the governmental Svefaktura project that was based on procurement needs. There were quite a lot of talks of invoicing theory and implementation issues. Different actors with different views made different approaches with different success rates.

It also included the contacts with XBRL by the co-operation with XBRL Sweden and the basic XBRL ideas (advocated/driven by FAR the audits SRF the accountant's organisations in Sweden). SIE also had at that time tight contacts with the Swedish national archive (riksarkivet) in digital archive topics and M. Gandy about user views.

UN/CEFACT had its 11th Forum in Stockholm Sweden in September 2007. We meet TBG12 Accounting group and contacts were made and kept for a few years. But most of all it was a very good opportunity to meet and get the views from many organisations and people would wide in various parts of business administration. Alphabet also visited the 14th UN/CEFACT Forum in Rome in 2009 where a progressive contact with Gianluca Garbellotto XBRL/GL International was made.

At the yearly meeting 2008 of SIE the organisation stated:

1. We will not become a formalised standard organisation
2. We are here to maintain the present successful SIE4 format
3. We will not involve in any matters except the SIE file format
4. We have no intensions of having any international contacts
5. We will not take any notice of any needs outside Sweden
6. We will not have any interactivity with governmental processes and the technical implementation of them

Jan Bergström and Alphabet AB did only agree to the first point and ended the secretary commission for the SIE board and left it.

Since Alphabet AB has been working on the Rosetta objectives above, using the experience from the work for the SIE group organisation.

## 5. How did the SIE4 format become so successful?

The core has been understanding why SIE4 made such a success and Alphabet have identified it to be the methods rather than file format.

The major reason official or wannabe standard organisations haven't had the same success is the ban on including methods in the work. No other organisational body has taken the methoding responsibility on top of the standard bodies basic work leading to confusion and no firm guidance in how to use formats. SIE is a user/vendor interest group and as such made both file format and methoding.

Most of the used file formats in the computer world have its origins in proprietary parties development projects (often in customer ordered missions). Some have later become official standard file formats (using the proprietary work that the standard organisation itself can't do).

In computer business they who make things works are the successful.

## 6. Accounting interchange methods – What is that?

There are three general levels of Accounting interchange methods:

1. The GAAP used (specified in the files and can vary quite)
2. The tax legislation (the EU VAT directive)
3. The technical implementation issues of information interchange files

Without the methods wrong interpretation of the information will be the general problem. It shows in symptoms like sums are not correct or the accounting has a very unfamiliar look and presentation (makes it hard to understand).

### 6.1. Methods of the GAAP used

The GAAP (Generally Accepted Accounting Principles) are similar but different in every country, defined domestically by the Institute of Certified Public Accountants and Audits or/and governmental bodies (that often are the same).

Typical differences in GAAP are like in Sweden non-transparent pre-systems are the general rule and illegal in most other countries. It means the accounting only contains the periodic sums of the actions in salary systems, payables and receivables.

Other issues are like the Swedish GAAP (BFN) demands paper invoices to be stored despite they being scanned and interpreted by use of OCR. This because the government general persistent resisting the implementation of the EU digital signature directive (1999/93/EG), now the EU digital signature regulation (910/2014 to be implemented 2016). In Sweden there are a large number of invoice scanning service bureaus and provider of scanning systems for the paper invoice receiving entities. No digital signing, the scanning party can't verify its interpretations digitally, not fulfil the demands of the GAAP. Else the scanned invoices would be certified and no paper bales needed, and very soon all accounting would become completely digital.

The missing digital signature infrastructure makes the VAT directive article 233 impossible to implement (and the VAT directive article 233 is excluded in the Swedish VAT law). It is also one of the major factors that make true digital receipts in the daily store trade impossible to be implemented successfully, receipts can't be signed by the issuing company, the concept of digital signatures of legal persons. Huge sums are invested by large retail chains in delivering digital receipts to their registered customers, when the banks refuse and the government resists. But with no method of standard connection to accounting it will end up in very poor results. Only governments can initiate digital signature and digital receipt (huge future markets) infrastructures.

Digital signatures would be used by everyone many times every day in all part of trade. But the firm resistance from governments (the lack of drive by the present EU commission of its own good work) is the day.

The GAAP is declared in XBRL/GL and in the Rosetta format. Reading foreign accounting files knowledge in local GAAPs is essential.

## **6.2. Methods of the tax legislation (the EU VAT directive)**

The VAT legislation is the primary legislation of accounting in every country. Not harmonised to the VAT law, the tax reports are invalid and about every entity of accounting has to do tax reports.

The major topic is the definition of the invoice and the handling of invoices in accounting, as accounting and tax reporting tools and objects.

In Europe a common VAT directive that is the father of all its member states VAT legislation is a very detailed, commercially outstanding concise and specific common work. It means that an invoice (article 226) is (computer-) logically the same invoice in every member state.

We are using the SIE methods of the EU VAT directive.

## **6.3. Methods of the technical implementation issues**

The SIE specification is focused on the technical implementation of its file format (that actually could be any file format) and is much more important than the file format syntax itself.

This describes the internal technical logics of how the file format should be used. For instance SIE says that the balances should be the sum of the accounting vouchers entries with accounting vouchers header dates of the balance period and nothing else. That the Opening balance should be defined by using the SIE Opening balance tags.

There are quite a lot of variants in internal logics in different accounting software. So an accounting program that uses an accounting entry to define the Opening balances has to remove it and put the data into SIE Opening balance tags. Else receiving programs will not understand the content of the file.

We are using the SIE methods in the Rosetta Accounting.

## **7. The market of the common man is pushing**

We all are waiting for the digital signatures infra structure and the extended credit card POS protocol with a digital receipt link. The pressure in Sweden is now getting strong, hardly any is using cash, everybody uses smartphones. The common man wants to see digital receipts in the internet bank account statements clicking.

We are working towards the society of paperless accounting with digital corporate and tax reporting.

Alphabet Rosetta accounting is designed to meet this new digital world.

## **8. The SIE/Alphabet Rosetta accounting methods**

### **8.1. Generating balance reports**

Rosetta (SIE) says that the balances should be the sum of the account of the vouchers entries (transaction lines) with the accounting entry vouchers dates of the balance period.

If Object Balances are made by only accounting vouchers entries tagged with the object, should be counted.

Deprecated Voucher Entries are not to be included. It is recommended by Rosetta (SIE) the amount zero not to be noted.

*There are quite a lot of variants in internal logics in different accounting software. The job is to translate their logics into Rosetta (SIE), in making Rosetta (SIE) files.*

## **8.2. Opening balance representation**

Rosetta (SIE) says that the Opening balance should be defined by using the Rosetta (SIE) Opening balance tags and nothing else.

*If an accounting program uses an accounting entry to define the Opening balances have to remove it and put the data into the Rosetta (SIE) Opening and Forward balance tags of that Ledger (and Opening balance of next).*

## **8.3. Opening and Forward balances representation**

SIE allows that only one of Opening and Forward balances are documented in the file, but both are recommended.

Rosetta demands that both are being used in the documentation because each ledger might be broken out of an accounting file.

*Them both are essential to certify that Forward balances one year matches Opening balances next year. Not uncommon is that the closing of the accounting year, could be reported that year (as Forward balances) but not transferred to next year (Opening balances), and there is a mismatch. A mismatch that points on that other adjustments in the next years accounting are made to match the year closing work of the previous. For instance changes in payables and receivables.*

## **8.4. Period sums should be documented**

SIE supports monthly period sums and recommend them being used.

Rosetta supports monthly, quarterly, tertially, Semi-annual and annual period sums that are essential for accounting analyse.

Rosetta demand them being used in the documentation, the Rosetta XML files. In Charts/Voucher content files, balances are not expected.

## **8.5. File location**

SIE says that the SIE-files would be stored in the \SIE-folder, because it is essential for other programs and apps to find the common generic accounting files. Rosetta is in line with this demand.

- In DOS it means C:\SIE
- In Windows it is interpret as C:\Users\[User]\Documents\Sie
- Equivalent placing in other operating system environments

## **9. The SIE/Alphabet Rosetta accounting object methods**

SIE /Alphabet Rosetta accounting objects are a way of marking accounting voucher entries with object tags. This allows the accounting to have specific balances for those transactions that are tagged with the object and is very useful.

The SIE /Alphabet Rosetta accounting objects gives the user (and developer of accounting systems) a huge possibility to make standardised own implementations that any other program can read and interpret. A user can define as many objects the user likes.

The objects are defined under different dimensions. The Dimensions represents different types/themes of objects. A user can define as many dimensions the user likes.

A SIE /Alphabet Rosetta accounting voucher entry can be tagged with as many objects the user likes.

For each SIE /Alphabet Rosetta accounting object the accounting system is expected to generate its own balance reports.

The tagging could also be the basis of computer processing.

The advantage of the SIE /Alphabet Rosetta object accounting is that it is fully transparent accounting transactions. An analyser can go into the smallest details. And still it is completely user defined.

The Alphabet Rosetta object have an optional Entity information set because many objects are physical or legal persons with such data.

It also has the ability to bind external references by Url-addresses.

### 9.1. Technical presentation Objects

A user can register any dimension with an Id (a numeric digit) larger than 20. Any numeric is OK.

The following dimension numbers are standardised and reserved. It is for all applications knowledge of data interchange. This way applications handling cost centres (any of the standard dimension themes), but not objects, can convert its content smoothly, write and understand reading files. And under these dimensions the user can also create objects:

1 = Cost centre / result unit

2 = Cost object (is to be sub-dimension of 1)

3 = Point of sales

4 = Branch

5 = Inventory (not acknowledged by SIE, acknowledged by Rosetta)

6 = Project

7 = Employee (deprecated by SIE, acknowledged by Rosetta)

8 = Customer (deprecated by SIE, acknowledged by Rosetta)

9 = Supplier (deprecated by SIE, acknowledged by Rosetta)

10 = Invoice (deprecated by SIE, acknowledged by Rosetta)

11-19 = Reserved for future expansion of the standard.

20 = Available for any user.

Under each dimension the user can register any object and the Object Ids are an alphanumeric string of any length.

The Rosetta objects also include the dimension Id so the same C-structure, C++-class or Java method could be used for object definitions under the definition of the dimension (where the dimension Id is given by the position). But also be used in tagging balance reports for the object and tagging the accounting voucher entries. Then getting the dimension Id this way. (The Rosetta C-engine manages this because they are pointers and not substructures.)

## **9.2. Cost centre, Cost Objects, Projects etc**

In many companies special result accounting for Cost centre, Cost Objects, Projects are very popular and this is made in SIE /Alphabet Rosetta accounting by tagging the voucher entries with objects.

The application is expected to generate object balance reports.

## **9.3. Payables and Receivables**

The Invoice dimension is in general used for Payables and Receivables accounting by giving every outstanding invoice in the Payable and Receivable an object. Accounting it in the accounting voucher with the Payable or Receivable account and its object.

When paid, accounting it in the accounting voucher with the Payable or Receivable account and its object, again. So when paid the object balance should be zero. This way also partly paid/instalment invoices can be accounted for.

The Payables and Receivables application uses the object list of the invoice dimension to construct debts and demand lists for the user to handle.

The Payables and Receivables are kept over changes in accounting years with the object balance information and the object definition information. In Rosetta it points to the invoice itself as well.

Rosetta can keep the Voucher list (journal) in different Ledgers (accounting years) in the same file/in memory (and import more) to be able to track an object over several years. Applications with SIE-support usually don't expect a SIE-file to have any accounting Vouchers (Verification items) than the present year (0).

The Payables and Receivables transactions could also be tagged with the objects of the Customer and Supplier dimensions to get record of the Customer and Supplier balances, which usually is very interesting.

It can also be combined in the accounting Voucher Entry with the Objects for the Cost centre, Cost Objects, Projects etc dimensions to get a full picture. The user can build up full information webs this way.

## **9.4. Salaries and Reimbursements**

The Employee dimension could be used for Employee Salaries and Reimbursements by tagging accounting information with the objects of the employees.

Very often there is an employees clearing account where the salaries and reimbursements are accounted for each employee at the last day of the month, tagged with each employee's object. Then at payday the payments are the balance of the employee and the salary specification (a self-billed invoice) is constructed by each accounting system for that period and employee. It will be the verification of the pay Voucher.

The quantity accounting facilities of the SIE /Alphabet Rosetta accounting gives the system/user the opportunity to count for instance work hours, holiday days, reimbursement km or whatever.

## **9.5. Inventories**

For tangible assets and inventories can be accounted for using objects in the Inventory dimension tagging any inventory or equipment with an inventory object.

This can then be used for down-writings, amortisations, depreciation and revaluation and to keep track of the unit technical data (attached to the Rosetta Invoice).

The tangible and inventories objects are kept over Ledgers (changes in accounting years) with the object balance information and the object definition information. In Rosetta it points to the invoice itself as well.

### **9.6. Technical tagging**

In many accountings multiple accounts are used for the same use but of different rates. For instance outgoing VAT that should be reported by the different rates in the EU Vat directive taxable countries.

Here a VAT-rate dimensions could be created and objects of the different rates. The objects are tagged to every accounting transaction of the outgoing VAT account and the object balances of the period are the basis for the report (without technically having to use more than one outgoing VAT account).

### **9.7. Period and objects combined**

In many report situations period balances of objects are very useful.

### **9.8. Volatile balanced accounting series**

It is possible to split up an accounting generated by different applications (pre-systems, like receivables that in some occasions can be very big), that together forms the entire accounting.

Sometimes (but less common when computer power and memory is larger today than in the past) it can be too much data for the applications for smooth handling.

Then a design of internally balanced pre-system own accounting series, use can be made very smooth:

- Every pre-system, like payables and salaries accounting Voucher ID series is balanced with a balance accounting voucher of the series.
- Every period (month) a mirror of the balance Voucher is reported to the main accounting
- Then the accounting works with or without the pre-system series of vouchers

This way it is possible to have the Payables, Receivables, Salaries, Reimbursements and Inventories in separate accounting series and bring them in and out of the accounting.

### **9.9. Transparency or not in pre-system accounting**

There is a general and intensive debate in the SIE society among its members if the SIE should support transparent accounting for pre-systems or not. In fact the original SIE 3 and SIE4 do support transparent accounting.

SIE5 don't support transparency and deprecated a number of default dimensions. The majority wants to stop transparent pre-system accounting in Sweden, because they represent another logical accounting method, non-transparency. A method wanted to be commonly used. Pre-system accounting not made by accounting accounts transactions but as data records. Records represented in the SIE5.

One of the topics of the debate is if it should be usable outside Sweden. The issue is that non-transparent accounting is illegal in most

countries and the question is if SIE should support something that is not universally usable? This is why some dimensions are deprecated.

It is quite possible to convert SIE5 non-transparent accounting to SIE4 transparent accounting just defining a new series and the payable and receivable accounts, putting an object on each item.

Alphabet AB with the Rosetta Accounting sees the world market as users. Not wanting to touch non-transparent accounting.

## 10. The Rosetta accounting file format – specification

The Alphabet Rosetta (and SIE4) accounting file format can only handle one entity in one file, can't handle files with multiple entities.

The methods and regulations in the SIE4 specification ([www.sie.se](http://www.sie.se)) are the basic methodical ground and the true guarantee of SIE compatibility. All accounting and financial software sold in Sweden the last 25 years have SIE support in import and export SIE file content. Also from very large distinguished international software suppliers.

The general ideas from XBRL/GL on common report formats are taken care of with great appreciation and so all attempts on a paper free digital accounting. To a large extent we are waiting for SWIFT, the EU and governments to supply digital receipts (VAT directive article 226b and 233) with credit card purchase (one URL reference line extension of the present credit card POS protocol).

We also see private enterprises as the retail chain ICAs ambitious (but in vain) internal registered customers digital receipt project. We need digital invoicing and Alphabet is trying to supply format for accounting and general purpose (in a world full of closed user group and procurement eInvoice projects).

### The basic framework [AccountingType](#) and the [InvoiceType](#)

Alphabet Rosetta accounting file format have to major XML-specifications the [AccountingType](#) and the [InvoiceType](#). At present the [AccountingType](#) are implemented because at the moment the file format is used as a common information container displaying SIE and XBRL/GL file content. Meaning that parts of the [AccountingType](#) is not yet fully used. This document describes the [AccountingType](#).

In the future versions of the Alphabet Rosetta accounting programs will use the [InvoiceType](#). The [InvoiceType](#) will be described in future versions of this document.

### 10.1. The [AccountingType](#)

The [AccountingType](#) has two major parts, the first is a container of the file where the general information of the entity and the file content is stored at the beginning. And followed by the actual accounting data.

#### 1. The file header information

```
<xs:complexType name="AccountingType">
  <xs:sequence>
    <xs:element name="Content" type="rac:ContentType" minOccurs="0" maxOccurs="1"/> <!-- #SIETYP ->
    <xs:element name="Comment" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/><!--#PROSA ->
    <xs:element name="Gaap" type="xs:token" minOccurs="1" maxOccurs="1"/> <!-- #KPTYP ->
    <xs:element name="Currency" type="rac:Currency" minOccurs="1" maxOccurs="1"/> <!--#VALUTA >
    <xs:element name="DefaultLanguage" type="xs:language" minOccurs="0" maxOccurs="1"/>
    <xs:element name="LastEntryDate" type="xs:date" minOccurs="0" maxOccurs="1"/> <!-- #OMFATTN->
    <xs:element name="FileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Program" type="rac:ProgramInfoType" minOccurs="1" maxOccurs="1"/> <!-- #PROGRAM >
    <xs:element name="GeneratingUser" type="rac:OperatorInfoType" minOccurs="1" maxOccurs="1"/> <!-- #GEN -->
    <xs:element name="AuditApproved" type="rac:AttestType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Entity" type="rac:PartyType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Accountat" type="rac:PartyType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Audit" type="rac:PartyType" minOccurs="0" maxOccurs="1"/>
```

#### 2. The data information charts

```
<xs:element name="AccountsChart" type="rac:AccountsChartType" minOccurs="0" maxOccurs="1"/>
<xs:element name="ObjectsChart" type="rac:ObjectsChartType" minOccurs="0" maxOccurs="1"/>
<xs:element name="LedgersChart" type="rac:LedgersChartType" minOccurs="0" maxOccurs="1"/>
<xs:element name="IssuedInvoiceChart" type="rac:InvoiceChartType" minOccurs="0" maxOccurs="1"/>
<xs:element name="BankAccountsChart" type="rac:InvoiceChartType" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
```

The accounting data is stored in five containers:

1. [AccountsChart](#)
2. [ObjectsChart](#)
3. [LedgersChart](#)
4. [IssuedInvoiceChart](#)
5. [BankAccountsChart](#)

**The accounts chart** container represents all the accounts registered for this entity's accounting. Not only those used, but also the inactive.

Equivalent to the SIE #KONTO, #KTYP, #ENHET tags.

**The objects chart** container represents all the accounting objects and dimensions registered for this entity's accounting. Not only those used but also the inactive. These are used to make special accounting for cost centres, projects, salaries, invoices etc. It is a very smooth way from SIE to handle advanced accounting on the conditions of the user.

Equivalent to the SIE #DIM, #UNDERDIM, #OBJEKT tags.

**The ledgers chart** container represents the all the reports and the journal, accounting entry vouchers of the accounting year.

The ledgers chart container represents the all the reports and the journal, accounting entry vouchers of the accounting year.

The ledgers charts can be of many accounting years.

Equivalent to the SIE #IB, #UB, #OIB, #OUB, #RES, #PSALDO, #BUDGET, #VER, #TRANS tags.

**The Issued invoice chart** container represents the issued invoices of the entity. This chart is a demand in many GAAPs to be held as a register of Issued invoices. However the payable and receivable should be administrated as objects in dimensions of invoices. It is not implemented in any programs yet but will be in future releases.

**The bank account statement chart** container is to hold bank account statements in digital form to be a basis and reference to the accounting, payable and receivable and other clearing accounts within the accounting. It is not implemented in any programs yet but will be in future releases.

**The top level of a Rosetta file looks like this:**

```
<Accounting
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="urn:Rosetta"
xsi:schemaLocation="urn:Rosetta http://www.alphabet.se/Rosetta/Rosetta.xsd">
  <Content>Ledger</Content>
  <Gaap>BAS96</Gaap>
  <Currency>SEK</Currency>
  <DefaultLanguage>sv</DefaultLanguage>
  <Program version="1.000">Rosetta SIE-XBRL/GL Viewer</Program>
  <GeneratingUser date = "2018-12-03" >Jan</GeneratingUser>
  <Entity>
    <Name>Testbolaget SIE AB</Name>
    <Address>Box 100</Address>
    <PostalAddress>12345 STORSTADEN</PostalAddress>
    <EntityPhone>08-1245678</EntityPhone>
    <ProprietaryId>C:\SPCSADM\FTG2</ProprietaryId>
    <LegalRegId>551264-1241</LegalRegId>
    <Jurisdiction>SE</Jurisdiction>
    <VatRegId>551264-1241</VatRegId>
    <VatCurrency>SEK</VatCurrency>
  </Entity>
  <AccountsChart>
  <ObjectsChart>
  <LedgersChart>
</Accounting>
```

## The file header information

The header has some general information that is of important interest of someone reading the accounting file.

### 10.1.1. Accounting **Content** information

The content is a enumeration of the following content alternatives:

- **Charts** where only Accounts and Objects charts are included
- **Balances** where only Balance reports are included no vouchers
- **Ledger** is a full accounting of Balances and Vouchers (journal)
- **Vouchers** a list of vouchers to be imported to an accounting

These are equivalent to the SIE #SIETYP tag.

### 10.1.2. Accounting **Comment** information

This allows the producer of the file to write a number of messages to the receiver as a free information set.

The **Comment** is of **TextType** and allowed in multiple one for each language. This allows multilingual accounting where the texts can be generated for many languages and users of different language use can read the same accounting.

Equivalent to the SIE #PROSA tag.

### 10.1.3. Accounting **Gaap** information

The GAAP used for the accounting is both very important to understand the accounting and to get the accounting chart identity.

Equivalent to the SIE #KPTYP tag.

### 10.1.4. Accounting **Currency** information

The currency of the accounting can be declared here. Logically an entire accounting is of one currency and any other currency is treated like quantified accounting.

Equivalent to the SIE #VALUTA tag.

### 10.1.5. Accounting **DefaultLanguage** information

Even if the accounting file contains multiple languages in every part (so it could be readable in different languages) one language is the default language the accountant have been using doing accounting.

### 10.1.6. Accounting **LastEntryDate** information

Last entry day is used when the last entry isn't the last day of the accounting year and obviously the accounting year is not yet finished.

Equivalent to the SIE #OMFATTN tag.

### 10.1.7. Accounting **FileUrl** information

In many cases an accounting relates to something that is to be handled or referenced to. This is how to connect the accounting to its digital documents.

This can be a file name in the same folder/computer as the accounting file but also a pointer to a file/webpage in the internet.

Note that company registration should be referred to [FileUrl](#) in the Entity section and tax registration to [FileUrl](#) in the Ledger for that accounting year,

### 10.1.8. Accounting [Program](#) information

The file producing program is documented here with its version as xml attribute.

Equivalent to the SIE #PROGRAM tag.

### 10.1.9. Accounting [GeneratingUser](#) information

The file generating user is documented here.

The [OperatorInfoType](#) contains the file generating name with the date of action as the XML attribute.

Equivalent to the SIE #GEN tag.

### 10.1.10. Accounting [AuditApproved](#) information

Here when the EU digital signature eIdas is implemented audit approval of the accounting can be registered here.

### 10.1.11. Accounting [Entity](#) information

All data of the Entity is stored here.

Equivalent to the SIE #FNAMN, #ADRESS, #FTYP, #FNR, #ORGNR, #BKOD tags.

### 10.1.12. Accounting [Accountat](#) information

All data of the Accountant consultant is stored here.

### 10.1.13. Accounting [Audit](#) information

All data of the Audit is stored here.

## 10.2. The [AttestType](#)

The [AttestType](#) is to be able to store digital approvals of document and accounting when the PKI digital signature infrastructure finally exists.

```
<xs:complexType name="AttestType">
  <xs:sequence>
    <xs:element name="Attestant" type="rac:OperatorInfoType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="Note" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="PkiNote" type="xs:token" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="FileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

### 10.2.1. Attest [Attestant](#) information

This filed is about who the attestant is.

The [OperatorInfoType](#) contains the attestants name with the date of action as the XML attribute.

### 10.2.2. Attest [Note](#) information

The attestant can leave information messages here.

The [Note](#) is of [TextType](#) and allowed in multiple one for each language. This allows multilingual accounting where the texts can be generated for many languages and users of different language use can read the same accounting.

### 10.2.3. Attest [PkiNote](#) information

The field is used for digital approval codes.

### 10.2.4. Attest [FileUrl](#) information

This is for related external documents.

## 10.3. The [PartyType](#)

The [PartyType](#) is to hold the information of entities, the accounting entity, the accountant consultant, audit and invoice parties.

Equivalent to the SIE #FNAMN, ADDRESS, FNR, ORGNR, FTYP and BKOD tags.

```
<xs:complexType name="PartyType">
  <xs:sequence>
    <xs:element name="Name" type="xs:token" minOccurs="1" maxOccurs="1"/> <!-- #FNAMN-->
    <xs:element name="Address" type="xs:token" minOccurs="0" maxOccurs="unbounded"/> <!-- #ADDRESS-->
    <xs:element name="PostalCode" type="xs:token" minOccurs="0" maxOccurs="1"/> <!-- #ADDRESS-->
    <xs:element name="PostalAddress" type="xs:token" minOccurs="0" maxOccurs="1"/> <!-- #ADDRESS-->
    <xs:element name="PostalCountry" type="xs:token" minOccurs="0" maxOccurs="1"/> <!-- #ADDRESS-->
    <xs:element name="EntityPhone" type="rac:Phone" minOccurs="0" maxOccurs="1"/> <!-- #ADDRESS-->
    <xs:element name="EntityEmail" type="rac:Email" minOccurs="0" maxOccurs="1"/>
    <xs:element name="LogotypeFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="WebPage" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="CrmRef" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="FileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Note" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Contact" type="rac:ContactType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ProprietaryId" type="xs:token" minOccurs="0" maxOccurs="1"/> <!-- #FNR -->
    <xs:element name="LegalRegId" type="xs:token" minOccurs="0" maxOccurs="1"/> <!-- #ORGNR -->
    <xs:element name="Jurisdiction" type="rac:Country" minOccurs="0" maxOccurs="1"/>
    <xs:element name="EntityType" type="xs:token" minOccurs="0" maxOccurs="1"/> <!-- #FTYP -->
    <xs:element name="StatisticalClass" type="xs:token" minOccurs="0" maxOccurs="1"/> <!-- #BKOD -->
    <xs:element name="LegalFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="VatRegId" type="xs:token" minOccurs="0" maxOccurs="1"/>
    <xs:element name="VatCurrency" type="rac:Currency" minOccurs="0" maxOccurs="1"/>
    <xs:element name="CashAccounting" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
    <xs:element name="VatFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="IncomeTaxFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

### 10.3.1. Party [Address](#) information

The [Address](#) field is multiple in order to handle the collapse of the ISO and the Universal Postal Union UPU Standard S42: International postal address components and templates.

The Rosetta solution is allowing multiple [Address](#) fields. The layout in printing is handled by the application program.

Equivalent to the SIE #ADDRESS tag.

### 10.3.2. Party [FileUrl](#) information

This is for related external documents.

### 10.3.3. Party [ProprietaryId](#) information

This is related to internal system IDs (definitely not required). Equivalent to the SIE #FNR tag.

### 10.3.4. Party **LegalRegId** information

This is the registration ID for the company register and income tax..  
Equivalent to the SIE #ORGNR tag.

### 10.3.5. Party **Jurisdiction** information

Define what national law regulates the company and its accounting.

### 10.3.6. Party **EntityType** information

This is the company of the company registration.  
Equivalent to the SIE #FTYP tag.

### 10.3.7. Party **StatisticalClass** information

This information is usually required in the tax reporting, and the information is stored to support tax administrative applications.

Put the codes of your domestic authorities demands.

*Statistical Classification of Economic Activities in the European Community, commonly referred to as NACE. There are NACE-related domestic versions in many countries, in Sweden SNI.*

*There's a correspondence between NACE and United Nations' International Standard Industrial Classification of all Economic Activities. NACE is similar in function to the SIC (Standard Industrial Classification) and NAICS (North American Industry Classification System) systems.*

Equivalent to the SIE #BKOD tag.

### 10.3.8. Party **LegalFileUrl** information

The **LegalFileUrl** is for external documents (or in other files in the same folder or computer) containing company information, like official company registration documents. .

### 10.3.9. Party **VatRegId, VatCurrency and CashAccounting**

These are demands of information from the EU VAT directive1.

Not supported in the Rosetta XBRL/GL Taxonomy.

## 10.4. The **ContactType**

The **ContactType** is to be able to store personal contact information.

```
<xs:complexType name="ContactType">
  <xs:sequence>
    <xs:element name="Function" type="xs:token" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Name" type="xs:token" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Address" type="xs:token" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="PostalCode" type="xs:token" minOccurs="0" maxOccurs="1"/>
    <xs:element name="PostalAddress" type="xs:token" minOccurs="0" maxOccurs="1"/>
    <xs:element name="PostalCountry" type="xs:token" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Phone" type="rac:Phone" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Email" type="rac:Email" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="WebPage" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="SocialMedia" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ContactLog" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="CrmRef" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Note" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

## Accounts Chart

### 10.5. The `AccountType`

The `AccountType` is to hold the accounting account information.

```
<xs:complexType name="AccountType">
  <xs:sequence>
    <xs:element name="Id" type="rac:IdToken" minOccurs="1" maxOccurs="1"/> <!--#KONTO >
    <xs:element name="Name" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/> <!--#KONTO >
    <xs:element name="QuantityUnit" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/> <!--#ENHET >
    <xs:element name="InActive" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
    <xs:element name="FileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Type" type="rac:AccountTypeType" minOccurs="0" maxOccurs="1"/> <!--#KTYP->
    <xs:element name="MiniChart" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
    <xs:element name="AccountCode" type="rac:AccountCodeType" minOccurs="0" maxOccurs="unbounded"> <!--#SRU/TAXAR>
    <xs:element name="OtherGaapAccountId" type="rac:AccountCodeType" minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Depreciation" type="rac:DepreciationsType" minOccurs="0" maxOccurs="1">
  </xs:sequence>
</xs:complexType>
```

#### 10.5.1. Account `Id` information

The `Id` is an alphanumeric string of any content. (SIE only support numeric account IDs) but some GAAPs do have alphanumeric Ids.

#### 10.5.2. Account `Title` information

The `Title` is of `TextType` and allowed in multiple one for each language. This allows multilingual accounting where the texts can be generated for many languages and users of different language use can read the same accounting.

#### 10.5.3. Account `QuantityUnit` information

The `QuantityUnit` is of `TextType` and allowed in multiple one for each language. This allows multilingual accounting where the texts can be generated for many languages and users of different language use can read the same accounting.

The `QuantityUnit` is bound to the account and so the quantity sum for each account can be calculated.

Note that accounts in other currencies are accounted in the accounting currency and the account currency is defined by the `QuantityUnit`.

Equivalent to the SIE #ENHET tag.

#### 10.5.4. Account `InActive` information

Is for script and report generating optimisations.

Not supported in the Rosetta XBRL/GL Taxonomy.

#### 10.5.5. Account `FileUrl` information

In many cases accounting accounts represents a contract or an invoice that is to be handled or referenced to. This is how to connect the accounting accounts to its digital documents.

This can be a file name in the same folder/computer as the accounting file but also a pointer to a file/webpage in the internet.

Only one supported in the Rosetta XBRL/GL Taxonomy.

### 10.5.6. Account **Type** information

Is an enumerated datafield of the following types:

- Sum-up title Total
  - Sum-up title Balance
    - Sum-up title Activa
      - **Tangible** (SIE & XBRL/GL Assets)
      - **Stock** (SIE & XBRL/GL Assets)
      - **Receivables** (SIE & XBRL/GL Assets)
      - **Assets**
    - Sum-up title Passiva
      - **Equities** (SIE Liabilities)
      - **Liabilities**
  - Sum-up title Result
    - Sum-up title Profit
      - **Revenues**
      - **Supplies** (XBRL/GL Expenses), direct costs
    - Sum-up title Earning (including the sum of Profit)
      - **Expenses**
      - **Staff**
      - **LossGains**
  - Other
    - **Financial**
    - **Auxiliary**

### 10.5.7. Account **MiniChart** marker

Supporting account selection in programs (used in Chart of accounts).  
Used for the most frequent accounts suggested by the GAAP provider.  
Not supported in the Rosetta XBRL/GL Taxonomy.

### 10.5.8. Account **AccountCode** information

This is account coding support, for instance connecting digital income tax report needs like the SIE #SRU and #TAXAR.

An account can be tagged with multiple **AccountCode** that are separated by the account code content information

Not supported in the Rosetta XBRL/GL Taxonomy.

### 10.5.9. Account **OtherGaapAccountId** information

This is account coding support, converting of the chart of accounts to another GAAPs account chart numbering.

An account can be tagged with multiple **OtherGaapAccountId** that are separated by the account code content information.

Not supported in the Rosetta XBRL/GL Taxonomy.

### 10.5.10. Account **Depreciation** information

This is for information if the account should be periodically subject of **Depreciation** of its value content. For instance write down of machinery in chunk.

Not supported in the Rosetta XBRL/GL Taxonomy.

## 10.6. The AccountCodeType

The `AccountCodeType` to hold the accounting account coding information.

Equivalent to the SIE #SRU and #TAXAR tags.

Not supported in the Rosetta XBRL/GL Taxonomy.

```
<xs:complexType name="AccountCodeType">
  <xs:sequence>
    <xs:element name="CodeType" type="xs:token" minOccurs="1" maxOccurs="1"/>
    <xs:element name="Issuer" type="xs:token" minOccurs="0" maxOccurs="1"/>
    <xs:element name="ApplicableDomain" type="rac:Country" minOccurs="1" maxOccurs="unbounded"/>
    <xs:element name="ValidYear" type="rac:Year" minOccurs="1" maxOccurs="1"/>
    <xs:element name="Code" type="xs:token" minOccurs="1" maxOccurs="1"/>
    <xs:element name="Description" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

### 10.6.1. Account Code CodeType information

The `CodeType` is a token used for a specific account code coding. This is to be found in automated processes in searching for a specific code type. Like when conversion between different GAAPs account numbering or a specific tax report generation.

### 10.6.2. Account Code Code information

The `Code` is a token of the actual marker used for a specific account code coding.

### 10.6.3. Account Code ValidYear information

This is to allow multiple `CodeType` token for the same purpose for different years, like need for income tax report generation. .

### 10.6.4. Account Code ApplicableDomain information

This is to allow multiple `CodeType` token for completely different use in different countries,

### 10.6.5. Account Code Description information

The `Description` is of `TextType` and allowed in multiple one for each language. This allows multilingual accounting where the texts can be generated for many languages and users of different language use can read the same accounting.

## Object Chart

### 10.7. The DimensionType

The **DimensionType** are themes to hold accounting objects and in the SIE methodology objects are tokens possible to put on accounting entry vouchers as many as the user likes. It is also there for machine solutions to handle different kind of object handling.

Objects can be for instance an item like a machine for down payment or write downs and with the objects getting Outgoing and incoming balances for each object. This means it is possible to keep track over the years.

Objects can be an employee and the salary and reimbursement accounting for this individual.

Objects can be a cost centre, department or project that the accounting is interested in keeping track on individual results and costs.

Equivalent to the SIE #DIM tag.

```
<xs:complexType name="DimensionType">
  <xs:sequence>
    <xs:element name="Id" type="xs:positiveInteger" minOccurs="1" maxOccurs="1"/> <!-- #DIM -->
    <xs:element name="Name" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/><!-- #DIM -->
    <xs:element name="SuperDimension" type="rac:DimensionType" minOccurs="0" maxOccurs="1"/> <!-- #DIM -->
    <xs:element name="InActive" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Object" type="rac:ObjectType" minOccurs="0" maxOccurs="unbounded"> <!-- #DIM -->
  </xs:sequence>
</xs:complexType>
```

#### 10.7.1. Dimension Id information

The **Id** is numeric. Dimensions 1-20 are reserved for specific use.

#### 10.7.2. Dimension Title information

The **Title** is of **TextType** and allowed in multiple one for each language. This allows multilingual accounting where the texts can be generated for many languages and users of different language use can read the same accounting.

#### 10.7.3. Dimension InActive information

Is for script and report generating optimisations.

Not supported in the Rosetta XBRL/GL Taxonomy.

#### 10.7.4. Dimension SuperDimension information

In SIE it is allowed to have one sub/super-level of dimensions.

Equivalent to the SIE #UNDERDIM tag.

## 10.8. Dimension **ObjectType**

The **ObjectType** to hold the accounting object information.

Equivalent to the SIE #OBJEKT tag.

```
<xs:complexType name="ObjectType">
<xs:sequence>
<xs:element name="Dimension" type="rac:DimensionType" minOccurs="1" maxOccurs="1"/>
<xs:element name="Id" type="rac:IdToken" minOccurs="1" maxOccurs="1"/> <!-- #OBJEKT -->
<xs:element name="Name" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/> <!-- #OBJEKT -->
<xs:element name="InActive" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
<xs:element name="FileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="Party" type="rac:PartyType" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
```

### 10.8.1. Object **Dimension** information

In many occasions the object is pointed at in the Accounting C-engine to allow immediate information update and no multiple data containers of the same thing. It is important to be able to locate the dimension of an object in many such cases.

### 10.8.2. Object **Id** information

The **Id** is an alphanumeric string of any content.

### 10.8.3. Object **Title** information

The **Title** is of **TextType** and allowed in multiple one for each language. This allows multilingual accounting where the texts can be generated for many languages and users of different language use can read the same accounting.

### 10.8.4. Object **InActive** information

Is for script and report generating optimisations.

Not supported in the Rosetta XBRL/GL Taxonomy.

### 10.8.5. Object **FileUrl** information

In many cases objects represents a contract or an invoice that is to be handled or referenced to. This is how to connect the object to its digital documents.

This can be a file name in the same folder/computer as the accounting file but also a pointer to a file/webpage in the internet.

Not supported in the Rosetta XBRL/GL Taxonomy.

### 10.8.6. Object **Party** information

Many objects are vendors, suppliers, employees etc that are legal or physical persons with for instance addressing issues.

The **Party** container in the object definition is to keep such data.

Not supported in the Rosetta XBRL/GL Taxonomy.

## Ledgers Chart

### 10.9. The LedgerType

The [LedgerType](#) is the container of one accounting years Ledger. The content are:

- The Balance Reports
- Accounting Vouchers list, the journal.

The [LedgerType](#) must not include both for every accounting year.

```
<xs:complexType name="LedgerType">
  <xs:sequence>
    <xs:element name="Id" type="xs:nonPositiveInteger" minOccurs="1" maxOccurs="1"/> <!-- #RAR -->
    <xs:element name="YearStart" type="xs:date" minOccurs="0" maxOccurs="1"/> <!-- #RAR -->
    <xs:element name="YearEnd" type="xs:date" minOccurs="0" maxOccurs="1"/> <!-- #RAR -->
    <xs:element name="FileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Report" type="rac:BalanceReportType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Voucher" type="rac:VoucherType" minOccurs="0" maxOccurs="unbounded"/><!--#VER/#TRANS>
  </xs:sequence>
</xs:complexType>
```

#### 10.9.1. Ledger Id information

The ledgers are numbered from present 0 to last year -1 and the year before -2 etc. This is in order to keep track of the Ledger accounting years listing of the accounting.

Part of the SIE #RAR tag.

#### 10.9.2. Ledger YearStart and YearEnd information

These dates are in the XML date format and defines the first and last dates of the accounting year. SIE and Rosetta do not support time of the date in the accounting. All accountings are date related.

Part of the SIE #RAR tag.

#### 10.9.3. Ledger FileUrl information

This can be a file name in the same folder/computer as the accounting file but also a pointer to a file/webpage in the internet.

VAT and Tax registration and other documents relating to the accounting year should be pointed to [FileUrl](#), in the Ledger for that accounting year.

Not supported in the Rosetta XBRL/GL Taxonomy.

#### 10.9.4. Ledger Report information

A number of Balance Reports of the Ledger accounting year, represents the entire accounting followed by one report for each object.

Equivalent to the SIE #IB, UB, OIB, OUB, PBUDGET and PSALDO tags.

#### 10.9.5. Ledger Voucher information

The [Voucher](#) information represents the list of accounting vouchers with the accounting entries of the Ledger accounting year.

Equivalent to the SIE #VER, TRANS, BTRANS and RTRANS tags.

## 10.10. The BalanceReportType

The `BalanceReportType` is the container of one Balance reports of the Ledgers entire accounting (no Object is noted) or for an object. The content are:

- The object
- The account type aggregations tags
- The account balances

```
<xs:complexType name="BalanceReportType">
  <xs:sequence>
    <xs:element name="Object" type="rac:ObjectType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Tangible" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Stock" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Receivables" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Assets" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Active" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Equities" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Liabilities" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Passive" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Balance" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Revenues" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Supplies" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Profit" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Expenses" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Staff" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="LossGains" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Earning" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Financial" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Auxiliary" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Result" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Total" type="rac:BalanceTotalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="BalanceAccount" type="rac:BalanceAccountType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

### 10.10.1. Report Object information

If no `Object` is noted the report stands for the Ledgers entire accounting. If an object is noted the report is for that object.

*The object have the same object representation as in the Objects chart to be possible to make 3d-structure handling with pointers in C++ programs and applications. Meaning that if an objects content is changed, it is automatically changed in all other parts of the accounting. This because all parts relating to an object point to the object itself, and not contain the object information.*

*Writing the file only the dimension number and object Id are necessary and the object name is optionally recommended to be written in the Rosetta files. All other object data could be left for the Object chart.*

### 10.10.2. Report account groups balance total tags information

In SIE there are only balance sums on account/object/period level.

In Rosetta account groups aggregations sums are there to support slow XML scripts (like XSL/XPATH/xhtml) performing generating readable reports. Account groups aggregations used should be noted.

This is not needed in standard to work, but in an XML document serving as an financial report the sums of the different account groups are central in the financial report. The layout program should not perform the mathematical sum ups of the report, should be in the report itself.

### 10.10.3. Report **BalanceAccount** information

Contains the balance and period sums of each account.

### 10.11. The **BalanceTotalType**

The **BalanceTotalType** is the container of one Balance reports of the Ledgers entire accounting (no Object is noted) or for an object. The content are:

```
<xs:complexType name="BalanceTotalType">
  <xs:sequence>
    <xs:element name="Opening"           type="rac:Amount"           minOccurs="0" maxOccurs="1"/>
    <xs:element name="Forward"          type="rac:Amount"           minOccurs="0" maxOccurs="1"/>
    <xs:element name="Month"            type="rac:BalanceTotalPeriodType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Quarter"         type="rac:BalanceTotalPeriodType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Tertial"         type="rac:BalanceTotalPeriodType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Semi"            type="rac:BalanceTotalPeriodType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Annual"          type="rac:BalanceTotalPeriodType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The balance totals do not contain Quantity information because quantities can only be aggregated in account level, having different Quantity units.

#### 10.11.1. Total **Opening** balance information

This is the sum of the opening balances of the accounts with this account code.

#### 10.11.2. Total **Forward** balance information

This is the sum of the forward balances of the accounts with this account code.

#### 10.11.3. Total period **Month, Quarter, Tertial, Semi and Annual**

This is the sum of the period Month, Quarter, Tertial, Semi and Annual of the accounts with this account code.

The periods are related to the calendar year and is to make period reporting possible. Month is 1 month, Quarter is 3 month, Tertial is 4 month, Semi is 6 month and Annual is 12 month report periods.

If the bookkeeping year is from 1 January to the 31 December Annual is not needed to be noted. Annual is to give calendar annual reports of entities with broken accounting years.

For entities with broken accounting years also other than Annual some periods will be over two bookkeeping years and recommended is that the period report is stored in both accounting years it covers.

Only month periods are supported in the Rosetta XBRL/GL Taxonomy.

### 10.12. The **BalanceTotalPeriodType**

The **BalanceTotalPeriodType** is the container of one Balance reports of the Ledgers entire accounting (no Object is noted) or for an object. The content are:

```
<xs:complexType name="BalanceTotalPeriodType">
  <xs:sequence>
    <xs:element name="End"               type="xs:date"               minOccurs="1" maxOccurs="1"/>
    <xs:element name="Outcome"          type="rac:Amount"           minOccurs="0" maxOccurs="1"/>
    <xs:element name="Budget"           type="rac:Amount"           minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
```

### 10.12.1. Total period **End** information

This is the end date in XML date format of the period (and start date is dependent on the period length).

### 10.12.2. Total period **Outcome** sum information

This is the sum of the outcome sum of the accounts with this account code of this period.

### 10.12.3. Total period **Budget** sum information

This is the budget estimation of the outcome sum of the accounts with this account code of this period.

## 10.13. The **BalanceAccountType**

The **BalanceAccountType** is the container of one Balance reports of the Ledgers entire accounting (no Object is noted) or for an object. The content are:

```
<xs:complexType name="BalanceAccountType">
  <xs:sequence>
    <xs:element name="Account" type="rac:AccountType" minOccurs="1" maxOccurs="1"/> <!--IB/UB/OIB/OUB/P/BU>
    <xs:element name="Opening" type="rac:Amount" minOccurs="0" maxOccurs="1"/> <!--IB/#UB/#OIB/#OUB->
    <xs:element name="OpeningQuantity" type="xs:decimal" minOccurs="0" maxOccurs="1"/> <!--IB/#UB/#OIB/#OUB->
    <xs:element name="Forward" type="rac:Amount" minOccurs="0" maxOccurs="1"/> <!--IB/#UB/#OIB/#OUB->
    <xs:element name="ForwardQuantity" type="xs:decimal" minOccurs="0" maxOccurs="1"/> <!--IB/#UB/#OIB/#OUB->
    <xs:element name="Month" type="rac:BalanceAccountPeriodType" minOccurs="0" maxOccurs="unbounded"/> <!--PS/#BU>
    <xs:element name="Quarter" type="rac:BalanceAccountPeriodType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Tertial" type="rac:BalanceAccountPeriodType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Semi" type="rac:BalanceAccountPeriodType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Annual" type="rac:BalanceAccountPeriodType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

### 10.13.1. The **Account** information

The **Account** is mandatory.

*The account have the same account representation as in the Accounts chart to be possible to make 3d-structure handling with pointers in C++ programs and applications. Meaning that if an accounts content is changed, it is automatically changed in all other parts of the accounting. This because all parts relating to an account point to the account itself, and not contain the account information.*

*Writing the file only the dimension number and account Id are necessary and the account name is optionally recommended to be written in the Rosetta files. All other account data could be left for the Account chart.*

### 10.13.2. Account **Opening** balance information

This is the sum of the opening balances of the account.

### 10.13.3. Account **OpeningQuantity** balance information

This is the quantity sum of the opening balances of the account.

### 10.13.4. Account **Forward** balance information

This is the sum of the forward balances of the account.

### 10.13.5. Account **ForwardQuantity** balance information

This is the quantity sum of the forward balances of the account.

### 10.13.6. Account **Month, Quarter, Tertial, Semi, Annual**

This is the sum of the period Month, Quarter, Tertial, Semi and Annual of the account.

The periods are related to the calendar year and is to make period reporting possible. Month is 1 month, Quarter is 3 month, Tertial is 4 month, Semi is 6 month and Annual is 12 month report periods.

If the bookkeeping year is from 1 January to the 31 December Annual is not needed to be noted. Annual is to give calendar annual reports of entities with broken accounting years.

For entities with broken accounting years also other than Annual some periods will be over two bookkeeping years and recommended is that the period report is stored in both accounting years it covers.

## 10.14. The **BalanceAccountPeriodType**

The **BalanceAccountPeriodType** is the container of one Balance reports of the Ledgers entire accounting (no Object is noted) or for an object.

The content are:

```
<xs:complexType name="BalanceAccountPeriodType">
  <xs:sequence>
    <xs:element name="End" type="xs:date" minOccurs="1" maxOccurs="1"/> <!-- #PSALDO/#BUDGET -->
    <xs:element name="Outcome" type="rac:Amount" minOccurs="0" maxOccurs="1"/> <!-- #PSALDO -->
    <xs:element name="OutcomeQuantity" type="xs:decimal" minOccurs="0" maxOccurs="1"/> <!-- #PSALDO -->
    <xs:element name="Budget" type="rac:Amount" minOccurs="0" maxOccurs="1"/> <!-- #BUDGET -->
    <xs:element name="BudgetQuantity" type="xs:decimal" minOccurs="0" maxOccurs="1"/> <!-- #BUDGET -->
  </xs:sequence>
</xs:complexType>
```

### 10.14.1. Account period **End** information

This is the end date in XML date format of the period (and start date is dependent on the period length).

Equivalent to the SIE #PSALDO and #PBUDGET tags.

### 10.14.2. Account period **Outcome** sum information

This is the sum of the outcome sum of the account, of this period.

Equivalent to the SIE #PSALDO tag.

### 10.14.3. Account period **OutcomeQuantity** sum information

This is the quantity sum of the outcome sum of the account, of this period.

Equivalent to the SIE #PSALDO tag.

### 10.14.4. Account period **Budget** sum information

This is the budget estimation of the outcome, of this period.

Equivalent to the SIE #PBUDGET tag.

### 10.14.5. Account period **BudgetQuantity** sum information

This is the quantity budget estimation of the outcome, of this period.

Equivalent to the SIE #PBUDGET tag.

## 10.15. The VoucherType

The **VoucherType** is the container of the data of the Invoice specified in the EU VAT directive 2006/112/EC article 226. The Vouchers form the Journal of accounting entries. The VAT directive legally defines how the Invoice as an accounting Voucher should be handled to perform VAT reporting.

Equivalent to the SIE #VER tag.

The content of the header is:

```
<xs:complexType name="VoucherType">
  <xs:sequence>
    <xs:element name="Id"
               type="rac:SeriesIdType"          minOccurs="1" maxOccurs="1"/>      <!-- #VER -->
    <xs:element name="Date"
               type="xs:date"                  minOccurs="1" maxOccurs="1"/>      <!-- #VER -->
    <xs:element name="Title"
               type="rac:TextType"            minOccurs="0" maxOccurs="unbounded"/> <!-- #VER -->
    <xs:element name="Accountant"
               type="rac:OperatorInfoType"     minOccurs="0" maxOccurs="1"/>      <!-- #VER -->
    <xs:element name="Entry"
               type="rac:VoucherEntryType"     minOccurs="0" maxOccurs="unbounded"/> <!-- #VER -->
  </xs:sequence>
</xs:complexType>
```

### 10.15.1. The Id information

The **SeriesIdType** is mandatory for the **VoucherType**.

The **SeriesIdType** is defined from the demand by the EU VAT directive Article 226 p1 “a sequential number, based on one or more series, which uniquely identifies the invoice”.

The **SeriesIdType** contains the sequential number with the series identifier as the XML attribute.

### 10.15.2. The Date information

The **Date** is mandatory for the **VoucherType**.

This is the Voucher date in XML date format.

This Voucher date is the only date used in generating balance and period reports.

### 10.15.3. The Title information

The Title is of **TextType** and allowed in multiple one for each language. This allows multilingual accounting where the texts can be generated for many languages and users of different language use can read the same accounting.

### 10.15.4. The Accountant information

Many GAAPs require the registration of the user generating the accounting entries of a voucher.

The **OperatorInfoType** contains the accountants name with the date of action as the XML attribute.

### 10.15.5. The Entry information

A list of accounting voucher entries of the accounting voucher.

It is allowed with empty accounting Vouchers with no entry lines.

## 10.16. The `VoucherEntryType`

The `VoucherEntryType` is the container of an accounting Voucher entry.

Equivalent to the SIE #TRANS, #BTRANS, #RTRANS tags.

The content are:

```
<xs:complexType name="VoucherEntryType">
  <xs:sequence>
    <xs:element name="Deprecated" type="rac:OperatorInfoType" minOccurs="0" maxOccurs="1"/> <!-- #BTRANS-->
    <xs:element name="Amount" type="rac:Amount" minOccurs="0" maxOccurs="1"/> <!-- #TRANS -->
    <xs:element name="Quantity" type="xs:decimal" minOccurs="0" maxOccurs="1"/> <!-- #TRANS -->
    <xs:element name="Title" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/> <!-- #TRANS -->
    <xs:element name="AuxDate" type="xs:date" minOccurs="0" maxOccurs="1"/> <!-- #TRANS -->
    <xs:element name="Account" type="rac:AccountType" minOccurs="1" maxOccurs="1"/> <!-- #TRANS -->
    <xs:element name="Invoice" type="rac:InvoiceType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Accountant" type="rac:OperatorInfoType" minOccurs="0" maxOccurs="1"/> <!-- #TRANS/RTRANS>
    <xs:element name="Object" type="rac:ObjectType" minOccurs="0" maxOccurs="unbounded"/> <!-- #TRANS -->
  </xs:sequence>
</xs:complexType>
```

### 10.16.1. The `Deprecated` information

Many GAAPs require by depreciation of accounting voucher entries that the deprecating user is registered and the depreciation time.

The `OperatorInfoType` contains the accountants name with the date of action as the XML attribute.

When an accounting Voucher entry is deprecated it is not included in the balances and periodic sums of the accounting.

It is also recommended that it should be hidden displaying the accounting voucher except when Audit mode for examination of the accounting is enabled.

### 10.16.2. Account `Amount` information

The `Amount` is mandatory.

This is the Amount (in the accounting currency) of the accounting voucher entry.

### 10.16.3. Account `Quantity` information

This is the Quantity (in the accounts quantity units) of the accounting voucher entry.

Note that making an accounting entry for a transaction in a different currency than the accounting currency of this entity the currency amount can be recorded as a Quantity and its currency as the Quantity unit of an account. This is typical handling of currency bank accounts in other currencies than the accounting currency. The Accounting currency amount in the accounting currency should be noted in the Amount field.

### 10.16.4. The `Account` information

The `Account` is mandatory.

*The account have the same account representation as in the Accounts chart to be possible to make 3d-structure handling with pointers in C++ programs and applications. Meaning that if an accounts content is changed, it is automatically changed in all other parts of the accounting. This because all parts relating to an account point to the account itself, and not contain the account information.*

*Writing the file only the dimension number and account Id are necessary and the account name is optionally recommended to be written in the Rosetta files. All other account data could be left for the Account chart.*

#### **10.16.5. The AuxDate information**

This is the Auxiliary Voucher date in XML date format for notes.

This date is only for note and synchronisation purpose. It can be used to note the date of the real action, when an accounting entry is moved between two accounting years. It can be used in synchronisation with back account statements when there are different entry dates in the back accounts of a back account money transfer. One has to be selected as the Voucher date, but for synchronisation purpose the entry line of the other back account can be tagged with the AuxDate.

#### **10.16.6. The Accountant information**

Many GAAPs require the registration of the user generating the accounting entries of a voucher an each entry line.

The `OperatorInfoType` contains the accountants name with the date of action as the XML attribute.

#### **10.16.7. The Object information**

Multiple `Object` can be tagged in an accounting Voucher entry generating Object balance and periodic sums for the object. Applications can also filter accounting vouchers on Objects.

*The object have the same object representation as in the Objects chart to be possible to make 3d-structure handling with pointers in C++ programs and applications. Meaning that if an objects content is changed, it is automatically changed in all other parts of the accounting. This because all parts relating to an object point to the object itself, and not contain the object information.*

*Writing the file only the dimension number and object Id are necessary and the object name is optionally recommended to be written in the Rosetta files. All other object data could be left for the Object chart.*

#### **10.16.8. Url/ Invoice information**

The `Invoice` is a container to hold interpret data from invoices externally linked by an Url.

If no interpretation is made the link can be to just the Url, by:  
`Invoice-> OriginalInvoiceFileUrl`

One Url is supported in the Rosetta XBRL/GL Taxonomy.

## **11. The Rosetta digital Invoice file format – specification**

### **11.1. The interpreting and process ability issue**

#### **11.1.1. The origins of file formats, usability**

The basic problem with electronic invoice formats is that they are designed for one non-accounting purpose, like for procurement. The reason is that projects have mainly been funded only to solve one issue, like electronic procurement. Accounting transparency and integration has been seen as too complicated to deal with and not funded. The accounting operators (auditors, accountants and software manufacturers have seen changes as risks of losing market shares) have been reluctant in supporting electronic invoice projects, no funding, and in their view no paying customer.

The basic legal regulation of invoices is the Vat legislation (the EU Vat directive) and accounting is indeed dependent on the legal regulations. In fact an invoice is primarily an accounting issue and about nothing else. Most electronic invoicing formats and systems have poor VAT directive support and no integration solutions to accounting.

The Rosetta invoice is an integrated part of accounting,

#### **11.1.2. The computability issue**

To efficiently connect transparency to accounting and get process ability in computer programs a standard file format for accounting systems is needed.

The basic idea is that interpreting (OCR of scanned or other electronic formats) data from other invoices, should be collected to a, by the accounting system known, specific set of data (in C called data structures, in C++ called classes and Java called methods), representing the XML format (defined in an XSD). The “Alphabet XSD to program code converter” creates the OS-independent C-Engine converting to C data structures and functions (C++ classes) and Java methods, from a given XSD. This is a program writing a program also to avoid bugs in its commercial programs/apps. It produces C functions (C++ classes) and Java methods to read, write and transparently process the data of the XML in a (system) standardised manner.

This means that:

- Interpreting invoices (of OCR of scanned or other electronic formats) is an issue of moving the data into a common known data structure.
  - Conversion facilities can be made as piggyback-code (code section, class/method) in apps and programs, to pre-process the data into a common data structure.
- The common data structure can be processed in a (system) standardised manner
  - With full transparency with the accounting for processing the data
  - One united computer program/app user presentation of invoice data can be made regardless of its initial format

### 11.1.3. A common invoice file format

By designing the accounting invoice file format for a common use, it will be smoother to use the file format directly. This requires that the Invoice file format beside the accounting and computability demands also includes:

- Formalised procurement needs
- Civil law and payment needs, including need for factoring and invoice collection. The need of being accurate in legal courts.
- Customer needs of purchase information, use, repair, security and disposal information
- The information distribution need in Web-trade
- The support of the computer industry
- That it is smooth to use and easy to understand the raw XML content.

In the development of the Alphabet Rosetta (accounting) electronic Invoice format we have tried to take such interest to allow a common future use.

### 11.2. Theoretic basis of what is an Invoice

It is important to realise that a number of needs are of an invoice and nothing else. Computer processing and for accurate legal interpretation it is important to be able to use a general format instead of specific task format variants for needs like:

- Clearing and Bank account statements
- Vat/Tax reports (self-billed)
- Salary statements (self-billed)
- Payables (a Clearing account)
- Pay order (a Clearing account)
- Procurement documentation (price lists, the ordering process documentation and contracts are also invoices)

Them all invoice-documents like offers, pricelists, acceptances, pay requirements, account statements, pay orders in a large number of very specific fields. In many cases self-billed self-debit statements.

### 11.3. Formalised procurement needs

Governmental (including municipal) procurement is a very important topic because EU directives enforce free and fair governmental procurement in its member states, the EU commission have demands.

So far it is in this field one of the largest sections of investments in electronic document interchange are made. The second is in internal company and suppliers trade networks like in the car industry.

### 11.4. The need of being accurate in legal courts

The Western world have experienced 50 years (1945-1995) of the extreme result driven management models (decentralised expert decision making, the manager is the team leader (like in football where there are still absolute result responsibility and funding is an important sub issue but not the main topic)). It also included a result driven economy steering model. It all developed in the US war machine against Japan in the IIWW being adapted in the civil society.

This resulting in the un-parallel economic development from poverty to welfare societies. Methods of absolute result focus with financing as

a sub-topic in the projects as funding issues. There were absolute demands of scientific methods and accuracy based law.

We are living in the post result-driven welfare society. A society of boss-management (boss makes all decisions and don't know the topics, making very bad quality decisions), abolishment of scientific method demands, abolishment in ability to appeal illegal decisions, "who you are"-driven civil law performed as a heavy debit lawyers talk-show. A return to the pre-Voltaire society of 1750. *We are back to the Leibnitz world and that is why the politics is so unstable lately, electorate don't accept it. Politicians likes Leibnitz and electorate Voltaire. Politics do not supply the needs, everybody talks about all others but themselves.*

One of the missing political reforms of the welfare society era, was reforming the legal systems. In many countries like Sweden the laws general basis is from pre-Voltaire ages 1734, supporting the boss-society and illegal business. Typical issues that are extreme legal problems for business but very good for lawyers pay are the ability to nonsense counter debit rotten invoices, making Reverse burden of proof of nonsense. Court acceptance demanding reimbursements for damages not yet appeared, but stated might happen. Courts refusal of demanding the parties to prove their claims before forcing them to bargain a settlement, forcing the parties to double their claims. All is standard civil law practice in Sweden, This is the daily legal conditions for business making its pay demands for services/goods, on customer credit. On top of that appeal limitations that comes as a response of extreme inaccuracy in legal performance. Where appeals is the lawyer making the legal talk show from the beginning to the end, once again.

The normal performance in result driven systems are that appeals are made on specific topics of a court decision. Like the court decision did not handle a specific formality right or that the evaluation of a specific factor is wrong. For instance that the other party can't prove his claims, no damages been made yet, etc. The appealed decisions should of course be sent back to the lower lever, for correction of its verdict, and no new talk show. Legal payments should be each parties issue and not to be claimed, just making the court sessions un-proportional and making the lawyers non-faithful to their clients. Billing court costs on the other party is the main reason small parties always loose to the big. The big makes processes un-proportional and the court do not protect the weaker party, trapping the weak party. Courts are just not efficient enough, they are still in the year of 1734. Payment court tasks should be performed by email, and not in court sessions. For that accurate documentation of business processes are needed.

The computer development and especially the internet and the web-page system have made the old legal document structures as information systems very obviously out of date. Also we can't see the renewal in the new needs of the web, any development of the digital receipts on credit card debits (VAT directive 233), the implementation of digital signatures (Digital signature directive/ eIDAS). And there is a fears resistance among governments in digital reporting of any kind like working ours in computer managed collective transportation and taxis, list is without end. The generals are no longer are defence ministers, lawyers are state attorneys, and it optimises the working market of the legal talk-show, not accuracy. Lawyer education (judges) do not include scientific methods and study of cases to understand processes. Meaning it is very hard to communicate business processes.

We saw under the previous EU commission a large number of very important simplifying projects of accounting, company registration, no demands on company capital if companies do not need it (like todays

IPR-based industry like the software business and consulting) and smoother audit. The governmental interpretation of the EU word simplification as abolishment, is not what it meant, but modernisation and computer transparency. Supporting accounting in mobile phones from internet bank account statements, introduce digital receipts and signatures, demanding computerised accounting. The member states turned the EU processes into nothing, on demand of them living on the present paper society. Them claiming they represent their clients.

The basic problem is that if civil law is not efficient and accurate enough, it is getting worse, business suffers legal inaccuracy and things like customer credits and smooth business can't be performed. This has been for instance one of the main hampering factors of business behind the previous iron curtain, not good enough legal civil law infrastructure. However there are ways to meet bad legal civil law infrastructure. An electronic invoice format should support also economies with bad infrastructure.

There is a need of development of extreme accuracy in documentation, preventing the legal talk-show and easy understanding of processes. Else business can't collect their bills, not good enough electronic invoices don't do.

### **11.5. Customer needs of purchase information, use, repair, security and disposal information**

There are no other good ways of general manner of digitally communicating customer needs of information like purchase data, use, repair, security and disposal information, than the digital invoice.

A private user of an internet banking system should be able to double-click on a debit transaction line of the internet bank account statement. Doing that the electronic receipt should appear including links to the information of purchase information (guarantee cards, product id, serial numbers, product specifications, product sheets etc), use (manuals), repair (including software upgrade info), security (legal demand nobody reads else) and disposal information (legal demand nobody reads else). If the bank is not storing the transaction data forever and perform search facilities the data must be possible to download into the mobile phone or PC of the customer, for processing.

An accountant should be able to make the accounting in a mobile phone of the transaction in the internet bank and get the invoice data with all its information links imported into the accounting.

From the accounting the company should be able to retrieve this data from its administrative system and distribute it to the employee that need the information for the performance of his work. Every employee should have the right of the full immediate retrievable information of all the tools and good being used.

The user should get this information regardless of the language of the user (multilingual). And it should work in any computer system/OS.

### **11.6. The information distribution need in Web-trade**

Today product information in web-trade sites are bought by the web-trader from a 3<sup>rd</sup> party because there is no natural distribution system of digital product information texts, pictures, complete product sheets.

In general this is the same problem as the general customer need of purchase information above and the electronic invoice is the natural distribution facility of this by document pointers, web-links.

The web-trade is getting more and more global and there is a need for multi-lingual product information distribution.

The natural solution is that the producer distributes the product information with document pointers in the invoice that is reused in the entire supply chain.

### **11.7. The support of the computer industry**

An electronic invoice must be designed for smooth computer processing in programs and apps produced by the computer industry.

One of the most important topics here is that all information should be performed with the specific target of general use. That the information is always applicable to any kind of computer application system.

The user should get this information regardless of the users language understanding. Meaning that all information (the file formats) must support indefinite many different languages of all texts.

The multilingual need of information texts demands a solution with a general text production structure that will work, In general a producer of a product produces its own product information texts. Many products are clusters of parts and the producer as customer of parts should be able to reuse his suppliers information to his products.

One important IPR topic of it is, that it must be free for anyone to use.

### **11.8. UN/CEFACT documents**

The UN has been working for many years in digital document development in the UN/CEFACT organisation. These are documents are for customs, transport/distribution, agricultural regulations and most needs. UN/CEFACT (previously EDIFACT) started performing cross-border trade and customs procedures for the UN member states. This in the statement trade makes war unprofitable.

The work has been performed in the way that different skilled specialist have been participating in developing a large number of (eb)XML digital document templates (XSD). Most of them are very good work and accepted in the different business branches.

In two fields the documents are not immediately usable, accounting and invoice (in cooperation with the OASIS) because of who were there and from what perspective they were made.

#### **11.8.1. UN/CEFACT Accounting – failed international support**

The UN/CEFACT TBG12 (that Alphabet AB participated in) was controlled by the French/Belgian accountant organisation Edificas (in control over accounting in France and Belgium). XBRL/GL International were there but agreement couldn't be made. SIE was there, did not agree and quitted.

The UN/CEFACT specification was documenting the Edificas process model of accounting. Like the Danish/German project of banks taking over accounting failed to get international support Edificas accounting processes failed to get international support. Only Edificas uses it.

The UN/CEFACT TBG12/ Edificas accounting was agreed with all the other work groups (but question is if the other really did understand what that meant and the huge problem for the entire UN/CEFACT project of not getting accepted and integrated in the international accounting world (outside Edificas in France and Belgium)).

The UN/CEFACT TBG12/ Edificas accounting processing model is also agreed on in the central UN/CEFACT management as the way of performing accounting processes in interaction with the other work groups performances. The general UN/CEFACT token processing model is not accepted in the accounting world.

The UN/CEFACT TBG12/ Edificas accounting processing model is based on the accounting token, a fragment of information transported as piggyback in the administrative computer processes of other documents like the invoice. This requires accounting systems to be accounting token based and not document based as they traditionally have been, and as described as documents in the EU VAT directive.

The basic difference of the internationally accepted accounting processes is that the entire invoice and the bank account statement or cash receipt are the only documents to be accounted for, as described in the EU VAT directive. To perform the UN/CEFACT TBG12/ Edificas accounting processing model Alphabet AB believes the VAT directive must be rewritten. Edificas in France and Belgium are not EU VAT directive harmonised, from that point of view.

The XBRL/GL and the SIE4 accounting processing model are fully compliant to the EU VAT directive invoice and accounting procedures demanded. The SIE5 (SIE XML) is not compliant to the VAT directive, deliberately (in order to defend traditions) not being transparent in pre-systems like salaries, inventories, payables and receivables. But SIE5 is compliant to Swedish law allowing non-transparency in accounting. SIE5 is not known to be implemented by anyone.

The Alphabet Rosetta specification (Rosetta.xsd) is fully compliant to (designed after) the VAT directive in both accounting and invoice.

### **11.8.2. The UN/CEFACT invoice – The accounting topic**

There are three general problems with the UN/CEFACT cross border invoice and all its variants (in the procurement process):

- It is dependent on the UN/CEFACT TBG12/ Edificas accounting processing model, that is out of market, incompatible processing with the international accounting world.
- The participants of the UN/CEFACT cross border invoice work group were mainly supplied from people working on the EU electronic procurement project, and not accounting. Invoices are legally regulated primarily as an accounting object (EU VAT directive). The work group performed documentation for supplying only procurement needs, not regulated accounting.
- UN/CEFACT cross border invoice work group co-operated with the OASIS Organization for the Advancement of Structured Information Standards that is dominated by large industry civil law needs. There has never been any firm integration with accounting (and the EU VAT directive).

The UN/CEFACT cross border invoice and the OASIS UBL invoice and all the national/local adaptations (normally incompatible to each other but similar) are not fulfilling the basic needs in accounting systems.

But they can be used (as civil law documents) and referred to by a web-link in accounting and the layout of the Alphabet Rosetta accounting invoice. A web-link to the UN/CEFACT cross border invoice, the OASIS UBL invoice and all the national/local adaptations, in the Rosetta invoice. And data can be collected in the Rosetta invoice to be processed.

And the UN/CEFACT cross border invoice and the OASIS UBL invoice developing processes have learned us most of what we know about electronic invoicing.

### **11.8.3. The UN/CEFACT implementation issue**

The basic problem of UN/CEFACT is a ISO standard organisation and OASIS is a wannabe standard organisation and as such follow standard organisations procedures and rules.

The basic rule is not have anything to do with proprietary solutions, anything that practically works, the instructions how to make it work. So the members of the organisations/the work groups have to perform the instructions how to make it work, themselves and that is not compatible to each other. It means the standard organisation supplies a large part of the members project needs, but they do not close the project as something working. There are no standard invoice in practice, just a huge number of cousin invoice types. The committee work also makes it easier to re-use each other computer routines.

### **11.8.4. Use of other UN/CEFACT documents**

Other parts of the UN/CEFACT work are complemented by other organisational bodies like vendor or user societies. Swift is the international banking organisation making the UN/CEFACT banking documents works with a common procedure between banks. The custom unions and bilateral custom agreements describes how to use the electronic documents of UN/CEFACT so they are indeed applicable documents in a general manner.

The Rosetta invoice is using their documents.

## **11.9. XBRL**

The XBRL/GL handles electronic invoices as external information links and as such anything could be pointed at. This allows technically and legally digital voucher folders and complete digital accounting (no paper documentation). But the accounting do not understand its content.

The Rosetta model of having an accounting invoice defined collecting the data in it and then point to the external document. Rosetta allows the invoice data generally be processed within its domain. External invoices are translated into Rosetta data structures in pre-systems.

## **11.10. Invoice link - The Accounting Voucher or its entry**

One accounting theory topic is if the Invoice should be related to the accounting voucher or its entry transaction lines.

Each accounting voucher is supposed to be relating to one document, the voucher (the paper document in the voucher folder) that is numbered in the voucher series of the journal.

However, how about a credit card invoice? It will have itself as the general voucher, but each transaction line will have another invoice (receipt) but very unpractical to account for in many cases as separate accounting vouchers.

We can see the same with many types of clear account debits.

In short the accounting voucher entry transaction line must have the ability to link itself to a specific digital document, as XBRL/GL is doing and so the Rosetta accounting format.

## 11.11. EU VAT directive

The EU VAT directive is the essential basis of the Rosetta invoice and its documentation is important here.

### 11.11.1. EU VAT union invoice definition

Within the European Union Value Added Tax directive, Article 226 is a concise definition of invoices within the union member states.

Without prejudice to the particular provisions laid down in this Directive, only the following details are required for VAT purposes on invoices issued pursuant to Articles 220 and 221:

- (1) the date of issue.
- (2) a sequential number, based on one or more series, which uniquely identifies the invoice.
- (3) the VAT identification number referred to in Article 214 under which the taxable person supplied the goods or services.
- (4) the customer's VAT identification number, as referred to in Article 214, under which the customer received a supply of goods or services in respect of which he is liable for payment of VAT, or received a supply of goods as referred to in Article 138.
- (5) the full name and address of the taxable person and of the customer.
- (6) the quantity and nature of the goods supplied or the extent and nature of the services rendered.
- (7) the date on which the supply of goods or services was made or completed or the date on which the payment on account referred to in points (4) and (5) of Article 220 was made, in so far as that date can be determined and differs from the date of issue of the invoice.
- (7a) where the VAT becomes chargeable at the time when the payment is received in accordance with Article 66(b) and the right of deduction arises at the time the deductible tax becomes chargeable, the mention 'cash accounting'.
- (8) the taxable amount per rate or exemption, the unit price exclusive of VAT and any discounts or rebates if they are not included in the unit price.
- (9) the VAT rate applied.
- (10) the VAT amount payable, except where a special arrangement is applied under which, in accordance with this Directive, such a detail is excluded.
- (10a) where the customer receiving the supply issues the invoice (instead of the supplier) the mention 'self-billing'.
- (11) in the case of an exemption, reference to the applicable provision of this Directive, or to the corresponding national provision, or any other reference indicating that the supply of goods or services is exempt.
- (11a) where the customer is liable for the payment of the VAT, the mention 'reverse charge'.
- (12) in the case of the supply of a new means of transport made in accordance with the conditions specified in Article 138(1) and (2)(a), the characteristics as identified in point (b) of Article 2(2).
- (13) where the margin scheme for travel agents is applied, the mention 'margin scheme — travel agents'.
- (14) where one of the special arrangements applicable to second-hand goods, works of art, collectors' items and antiques is applied, the mention 'margin scheme — second-hand goods', 'margin scheme — works of art' or 'margin scheme — collector's items and antiques' respectively.
- (15) where the person liable for payment of VAT is a tax representative for the purposes of Article 204, the (VAT identification number, referred to in Article 214, of that tax representative, together with his full name and address.

### 11.11.2. EU VAT union receipt (simple invoice) definition

Article 226b of the EU VAT Directive is a concise definition of receipts within the union member states.

As regards simplified invoices issued pursuant to Article 220a and Article 221(1) and (2), Member States shall require at least the

following details:

- (a) the date of issue.
- (b) identification of the taxable person supplying the goods or services (VAT identification number).
- (c) identification of the type of goods or services supplied.
- (d) the VAT amount payable or the information needed to calculate it.
- (e) where the invoice issued is a document or message treated as an invoice pursuant to Article 219, specific and unambiguous reference to that initial invoice and the specific details which are being amended.

They may not require details on invoices other than those referred to in Articles 226, 227 and 230.

### 11.11.3. EU VAT union electronic invoice demands

The EU VAT directive have demands on the validity of invoices that mainly refers to the EU digital signature directive today eIDAS (that most government refuse to implement since 1999). It is of huge important to know the text of the VAT directive in constructing accounting administrative computer systems.

**Article 232 The use of an electronic invoice** shall be subject to acceptance by the recipient.

**Article 233 1. The authenticity of the origin**, the integrity of the content and the legibility of an invoice, whether on paper or in electronic form, shall be ensured from the point in time of issue until the end of the period for storage of the invoice.

Each taxable person shall determine the way to ensure the authenticity of the origin, the integrity of the content and the legibility of the invoice. This may be achieved by any business controls which create a reliable audit trail between an invoice and a supply of goods or services.

'Authenticity of the origin' means the assurance of the identity of the supplier or the issuer of the invoice.

'Integrity of the content' means that the content required according to this Directive has not been altered.

**2. Other than by way** of the type of business controls described in paragraph 1, the following are examples of technologies that ensure the authenticity of the origin and the integrity of the content of an electronic invoice:

**(a) an advanced electronic signature** within the meaning of point (2) of Article 2 of Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures ( 1 ), based on a qualified certificate and created by a secure signature creation device, within the meaning of points (6) and (10) of Article 2 of Directive 1999/93/EC.

**(b) electronic data interchange (EDI)**, as defined in Article 2 of Annex 1 to Commission Recommendation 1994/820/EC of 19 October 1994 relating to the legal aspects of electronic data interchange ( 1 ), where the agreement relating to the exchange provides for the use of procedures guaranteeing the authenticity of the origin and integrity of the data.

## 11.12. The basis for an accounting electronic invoice format

The objectives of the Rosetta invoice have been:

- **A good accounting container** to keep interpreted (OCR of scanned or other electronic formats) data from other invoices
- **Being a good electronic invoice format for daily civil legal use** with full transparency with the accounting.
  - Presenting a easy to understand and easy readable basic XML-data invoice of accounting, audit and business administration
  - Harmonise with the EU VAT directive 2006/112/EG, especially the specification and logical concept of:
    - The Invoice (article 230)

- The Receipt (article 226b)
- The Digital invoice (article 233)
- The conceptual idea and logics that the following also are invoice implementations:
  - Clearing and Bank account statements
  - Vat/Tax reports (self-billed)
  - Salary statements (self-billed)
  - Payables (a Clearing account)
  - Pay order (a Clearing account)
  - Procurement documentation (price lists, the ordering process documentation and contracts)
- The VAT directive is such a good basic documentation that it is expected to be copied also in non-EU member states. Copying/relating to it has major advantages.
- The general needs of the commercial trade, especially web-based atomised trade systems
  - Product type information
  - Product serial number information
  - Distribution of product information/digital product sheets (for web-trade sites)
  - Support for factoring and Invoice collection
  - Support of procurement procedures
  - Making supplies transparently traceable
  - Supporting automated supply chains (ability to design automated web trade administrative systems)
  - Safety and environmental handling information to be distributed in the supply chain handling
  - Transparency to the nuts and bolts with logistics and integration of logistics and accounting inventories.
- The general need of customers getting purchase information retrievable from accounting and business documentation. Supply of digital invoices and digital receipts with high information standards. Being able to retrieve:
  - Product type information
  - Product serial number information
  - Digital product sheets
  - Guarantee sheets
  - User registration information
  - Manuals and user instruction sheet
  - Technical support documents
  - Faulty product handling
  - Repairs and maintenance information
  - Safety and environmental handling information in use of the products
  - Recycling information
- Multiple language support:
  - For international trade (Multilingual invoices)
  - For countries with multiple language markets (multilingual invoice texts and product information)
  - For international business interaction (being able to read accounting over the borders)

### 11.13. The InvoiceType

The `InvoiceType` is an accounting document that also serves many other means, the `Purpose` field defines what. The Rosetta invoice is designed to server them all.

In the transformation phase of accounting it is used as an information container of the voucher referred to by the accounting voucher entry.

Data from the original invoice (Scanned OCR or electronic formats) are stored in the Rosetta invoice container to be possible for computers to process.

The content are:

```
<xs:complexType name="InvoiceType">
  <xs:sequence>
    <xs:element name="Purpose" type="rac:InvoiceTypeType" minOccurs="0" maxOccurs="1"/> EUVAT 226p10a
    <xs:element name="LegalLanguage" type="xs:language" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Deprecated" type="xs:date" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Title" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/> EUVAT 226p6
    <xs:element name="InvoiceId" type="rac:SeriesIdType" minOccurs="0" maxOccurs="1"/> EUVAT 226p2
    <xs:element name="Id" type="xs:token" minOccurs="0" maxOccurs="1"/>
    <xs:element name="PayableId" type="rac:SeriesIdType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="PayObject" type="rac:ObjectType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="PaidDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
    <xs:element name="DateOfIssue" type="xs:date" minOccurs="0" maxOccurs="1"/> EUVAT 226p1
    <xs:element name="DateOfArrival" type="xs:date" minOccurs="0" maxOccurs="1"/>
    <xs:element name="DueDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
    <xs:element name="LatePayInterestRate" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
    <xs:element name="CreditLimit" type="rac:AmountCurrencyType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="PayReference" type="xs:token" minOccurs="0" maxOccurs="1"/>
    <xs:element name="PayCurrency" type="rac:Currency" minOccurs="0" maxOccurs="1"/>
    <xs:element name="PayAmount" type="rac:Amount" minOccurs="0" maxOccurs="1"/>
    <xs:element name="VatAmountPayCurrency" type="rac:Amount" minOccurs="0" maxOccurs="1"/> EUVAT 226p10
    <xs:element name="NetAmountPayCurrency" type="rac:Amount" minOccurs="0" maxOccurs="1"/> EUVAT 226p8
    <xs:element name="VatAmountTaxCurrency" type="rac:Amount" minOccurs="0" maxOccurs="1"/> EUVAT 226p10
    <xs:element name="NetAmountTaxCurrency" type="rac:Amount" minOccurs="0" maxOccurs="1"/>
    <xs:element name="BuyersNote" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="SellersNote" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="FactoringNote" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="CollectingNote" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="BailiffNote" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="OriginalInvoiceFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="RelatedDocumentFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Issuer" type="rac:AttestType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="PayApproved" type="rac:AttestType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="PayOrderd" type="rac:AttestType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="InvoiceEntry" type="rac:InvoiceEntryType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="InvoiceEntry" type="rac:InvoiceEntryType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="AllowanceCharge" type="rac:InvoiceEntryType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="LogisticsServiceCharge" type="rac:InvoiceEntryType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="CollectionCharge" type="rac:InvoiceEntryType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Discounts" type="rac:InvoiceEntryType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Buyer" type="rac:PartyType" minOccurs="0" maxOccurs="1"/> EUVAT 226p4,5
    <xs:element name="Seller" type="rac:PartyType" minOccurs="0" maxOccurs="1"/> EUVAT 226p3,5
    <xs:element name="TaxRepresentative" type="rac:PartyType" minOccurs="0" maxOccurs="1"/> EUVAT 226p15
    <xs:element name="Factor" type="rac:PartyType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Collector" type="rac:PartyType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Bailiff" type="rac:PartyType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ReferenceAccounting" type="rac:SeriesIdType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="AmortisationSchedule" type="rac:AmortisationScheduleType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Pay" type="rac:PayType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

### 11.13.1. The Purpose information

As described above in 10.2 Theoretic basis of what is an Invoice, an invoice could be for a large number of use and the tells for what use (what the reader is expected from it). It is a enumerated list of choice:

- Simple
- Receipt
- Invoice
- Factoring
- Credit
- Reminder
- Collection
- Bailiff
- SelfDebit
- SelfCredit
- Salary
- Reimbursement
- PayOrder
- VAT
- PayrollTax
- INC
- TAX
- BankStatements
- Clearing
- PriceList
- Tender
- Offer
- Order
- Response
- Contract
- Proforma
- Depreciation
- Note

### 11.13.2. The Deprecated information

If the invoice is deprecated, credit invoices are often sent, then deprecated is marked on the invoice to recognise its credit state.

### 11.13.3. The InvoiceId information

Noted by the issuing party, by issuing the invoice.

This is Invoice Id in the VAT directive meaning of Invoice serial Id.

We recommend that it is using the same as object ID in the accounting with dimension of invoice and the series and Id as object ID.

### 11.13.4. The Id information

Noted by the issuing party, by issuing the document.

This token-id is used to identify documents like contracts, offers, orders, bank accounts etc. It is not an Invoice Id in the VAT directive meaning. It is a civil law token of identity of the document.

### 11.13.5. The PayableId information

Noted by the paying party, in back office.

If the paying party issues internal payable numbering series, it should be noted here. We recommend that it is using the same as object ID in

the accounting with dimension of invoice and the series and Id as object ID.

#### **11.13.6. The PayObject information**

Noted by the both parties, in back office.

Note of the internal accounting Object. Note of Object ID and the dimension ID (Invoice), supplier and customer objects is also suggested to be noted.

#### **11.13.7. The PayReference information**

Noted by the issuing party.

Is an identification token of the claim to be noted on the payment.

#### **11.13.8. The PayCurrency information**

Noted by the issuing party.

Is the currency the payment is expected to be delivered in. In the accounting this is noted in the invoice data of the accounting voucher and is only handled by the payable pay order function.

#### **11.13.9. The PayAmount information**

Noted by the issuing party, the civil law pay claim in pay currency.

#### **11.13.10. The BuyersNote, SellersNote, FactoringNote, CollectingNote, and BailiffNote information**

To put notes in both at issuing and later in back office.

#### **11.13.11. The Issuer information**

Noted by the issuing party, by issuing.

The VAT directive article 233 demands digital invoices to be digitally signed. Even not implemented the issuer information can be a good information for communication between the parties.

#### **11.13.12. The PayApproved and PayOrderd information**

Noted by the paying party, in back office.

Approval in the payable process. Note that the **ReceivingApproved** is in the Invoice entry level (because different goods might be received by different officials). When delivery is acknowledged **PayApproved** can be issued. The **PayOrderd** is made by the accountant when pay order Is sent.

It also allows internal PKI digital signatures to be used even though the society is not supplying an digital signature infra structure.

#### **11.13.13. The InvoiceEntry, AllowanceCharge, LogisticsServiceCharge, CollectionCharge, and Discounts information**

Invoice entries can be separated according to the debit type.

#### 11.13.14. The Buyer, Seller, TaxRepresentative, Factor, Collector, and Bailiff information

Invoice party entity information.

#### 11.13.15. The ReferenceAccounting information

Used by the administrative computer systems.

This to allow the computer system to point at the accounting voucher.

#### 11.13.16. The AmortisationSchedule information

Noted by any party.

A note if the payment follows an amortisation schedule, like payments of a loan.

#### 11.13.17. The Pay information

Noted by the issuing party, by issue.

This is the actual claim of payments of the invoice amount.

### 11.14. The InvoiceEntryType

The `InvoiceEntryType` is the specification of the invoice (demanded by the VAT directive article 226 p6, 7, 8, 9, 10, 11, 11a, 12, 13 and 14).

The content are:

```
<xs:complexType name="InvoiceEntryType">
<xs:sequence>
<xs:element name="Quantity" type="xs:decimal" minOccurs="0" maxOccurs="1"/> EUVAT 226p6
<xs:element name="Description" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/> EUVAT 226p6
<xs:element name="TradeItemData" type="rac:TradeItemDataType" minOccurs="0" maxOccurs="unbounded"/> EUVAT 226p6
<xs:element name="Quota" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
<xs:element name="QuotaBalance" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
<xs:element name="QuantityUnit" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="QuotaAgreedUnitNetAmount" type="rac:AmountCurrencyType" minOccurs="0" maxOccurs="1"/>
<xs:element name="Id" type="xs:token" minOccurs="0" maxOccurs="1"/>
<xs:element name="InvoiceId" type="rac:SeriesIdType" minOccurs="0" maxOccurs="1"/>
<xs:element name="AccountingDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
<xs:element name="AvailabilityDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
<xs:element name="RateOfIntrestDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
<xs:element name="SalesReference" type="xs:token" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="SalesObject" type="rac:ObjectType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="CostReference" type="xs:token" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="CostObject" type="rac:ObjectType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="Note" type="rac:TextType" minOccurs="1" maxOccurs="unbounded"/>
<xs:element name="DeliveryDate" type="xs:date" minOccurs="0" maxOccurs="1"/> EUVAT 226p7
<xs:element name="DeliveryNote" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="LocationNote" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="Location" type="xs:token" minOccurs="0" maxOccurs="1"/>
<xs:element name="DispatchFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="OrderId" type="xs:token" minOccurs="0" maxOccurs="1"/>
<xs:element name="ContractId" type="xs:token" minOccurs="0" maxOccurs="1"/>
<xs:element name="PriceListId" type="xs:token" minOccurs="0" maxOccurs="1"/>
<xs:element name="OrderResponseFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="SellersNoteToBuyer" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="SuggestedBasAccount" type="xs:token" minOccurs="0" maxOccurs="1"/>
<xs:element name="UnitNetAmount" type="rac:Amount" minOccurs="0" maxOccurs="1"/> EUVAT 226p8
<xs:element name="NetAmount" type="rac:Amount" minOccurs="1" maxOccurs="1"/> EUVAT 226p8
<xs:element name="VatReduced" type="rac:VatReducedType" minOccurs="0" maxOccurs="1"/> EUVAT 226p8
<xs:element name="VatSpecial" type="rac:VatSpecialType" minOccurs="0" maxOccurs="1"/> EUV 226p8,11,11a,13,14
<xs:element name="VehicleVatDocFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/> EUVAT 226p12
<xs:element name="VatRate" type="xs:decimal" minOccurs="0" maxOccurs="1"/> EUVAT 226p9
<xs:element name="VatAmountTaxCurrency" type="rac:Amount" minOccurs="0" maxOccurs="1"/> EUVAT 226p10
<xs:element name="PayAmount" type="rac:Amount" minOccurs="1" maxOccurs="1"/>
<xs:element name="ReceivingApproved" type="rac:AttestType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="StackedInvoice" type="rac:InvoiceType" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
```

### 11.14.1. The Basic Invoice section

The basic demand of an invoice entry is the item description.

```
<xs:complexType name="InvoiceEntryType">
<xs:sequence>
<xs:element name="Quantity" type="xs:decimal" minOccurs="0" maxOccurs="1"/> EUVAT 226p6
<xs:element name="Description" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/> EUVAT 226p6
<xs:element name="TradeItemData" type="rac:TradeItemDataType" minOccurs="0" maxOccurs="unbounded"/> EUVAT 226p6
<xs:element name="Quota" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
<xs:element name="QuotaBalance" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
<xs:element name="QuantityUnit" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/>
```

Must be used (according to EU VAT directive) for commercial invoices.

### 11.14.2. The Procurement section

The procurement quota information can be used in contracts as prime entry information and in commercial invoices as references to quotas.

```
<xs:complexType name="InvoiceEntryType">
<xs:sequence>
<xs:element name="Quantity" type="xs:decimal" minOccurs="0" maxOccurs="1"/> EUVAT 226p6
<xs:element name="Description" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/> EUVAT 226p6
<xs:element name="TradeItemData" type="rac:TradeItemDataType" minOccurs="0" maxOccurs="unbounded"/> EUVAT 226p6
<xs:element name="Quota" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
<xs:element name="QuotaBalance" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
<xs:element name="QuantityUnit" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="QuotaAgreedUnitNetAmount" type="rac:AmountCurrencyType" minOccurs="0" maxOccurs="1"/>
<xs:element name="Id" type="xs:token" minOccurs="0" maxOccurs="1"/>
<xs:element name="InvoiceId" type="rac:SeriesIdType" minOccurs="0" maxOccurs="1"/>
<xs:element name="AccountingDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
<xs:element name="AvailabilityDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
```

that the deprecating user is registered and the deprecation time.

### 11.14.3. The Bank/Clearing statement section

In Bank and clearing accounts transaction line Ids are used to separate transactions from each other (can be multiple the same).

Also the information on interest rates and dates are used here.

```
<xs:complexType name="InvoiceEntryType">
<xs:sequence>
<xs:element name="Quantity" type="xs:decimal" minOccurs="0" maxOccurs="1"/> EUVAT 226p6
...
<xs:element name="QuantityUnit" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="QuotaAgreedUnitNetAmount" type="rac:AmountCurrencyType" minOccurs="0" maxOccurs="1"/>
<xs:element name="Id" type="xs:token" minOccurs="0" maxOccurs="1"/>
<xs:element name="InvoiceId" type="rac:SeriesIdType" minOccurs="0" maxOccurs="1"/>
<xs:element name="AccountingDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
<xs:element name="AvailabilityDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
<xs:element name="RateOfIntrestDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
<xs:element name="SalesReference" type="xs:token" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="SalesObject" type="rac:ObjectType" minOccurs="0" maxOccurs="unbounded"/>
```

that the deprecating user is registered and the deprecation time.

#### 11.14.4. The Trade data section

In trade of goods and services a quite some information is needed to describe the trade information.

```
<xs:complexType name="InvoiceEntryType">
<xs:sequence>
<xs:element name="Quantity" type="xs:decimal" minOccurs="0" maxOccurs="1"/> EUVAT 226p6
...
<xs:element name="AvailabilityDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
<xs:element name="RateOfIntrestDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
<xs:element name="SalesReference" type="xs:token" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="SalesObject" type="rac:ObjectType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="CostReference" type="xs:token" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="CostObject" type="rac:ObjectType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="Note" type="rac:TextType" minOccurs="1" maxOccurs="unbounded"/>
<xs:element name="DeliveryDate" type="xs:date" minOccurs="0" maxOccurs="1"/> EUVAT 226p7
<xs:element name="DeliveryNote" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="LocationNote" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="Location" type="xs:token" minOccurs="0" maxOccurs="1"/>
<xs:element name="DispatchFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="OrderId" type="xs:token" minOccurs="0" maxOccurs="1"/>
<xs:element name="ContractId" type="xs:token" minOccurs="0" maxOccurs="1"/>
<xs:element name="PriceListId" type="xs:token" minOccurs="0" maxOccurs="1"/>
<xs:element name="OrderResponseFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="SellersNoteToBuyer" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="SuggestedBasAccount" type="xs:token" minOccurs="0" maxOccurs="1"/>
<xs:element name="UnitNetAmount" type="rac:Amount" minOccurs="0" maxOccurs="1"/> EUVAT 226p8
<xs:element name="NetAmount" type="rac:Amount" minOccurs="1" maxOccurs="1"/> EUVAT 226p8
```

that the deprecating user is registered and the deprecation time.

##### 11.14.4.1. The SuggestedBasAccount information

The [SuggestedBasAccount](#) is a suggestion of bookkeeping account in BAS mini-chart chart of accounts account ID to simplify accounting after invoice entry information. It is supporting using tools like mobile phones as accounting computer. It is suggesting an account when it is of a common nature.

#### 11.14.5. The Money section

The money section is used to describe the monetary information,

```
<xs:complexType name="InvoiceEntryType">
<xs:sequence>
<xs:element name="Quantity" type="xs:decimal" minOccurs="0" maxOccurs="1"/> EUVAT 226p6
...
<xs:element name="SellersNoteToBuyer" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="SuggestedBasAccount" type="xs:token" minOccurs="0" maxOccurs="1"/>
<xs:element name="UnitNetAmount" type="rac:Amount" minOccurs="0" maxOccurs="1"/> EUVAT 226p8
<xs:element name="NetAmount" type="rac:Amount" minOccurs="1" maxOccurs="1"/> EUVAT 226p8
<xs:element name="VatReduced" type="rac:VatReducedType" minOccurs="0" maxOccurs="1"/> EUVAT 226p8
<xs:element name="VatSpecial" type="rac:VatSpecialType" minOccurs="0" maxOccurs="1"/> EUV 226p8,11,11a,13,14
<xs:element name="VehicleVatDocFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/> EUVAT 226p12
<xs:element name="VatRate" type="xs:decimal" minOccurs="0" maxOccurs="1"/> EUVAT 226p9
<xs:element name="VatAmountTaxCurrency" type="rac:Amount" minOccurs="0" maxOccurs="1"/> EUVAT 226p10
<xs:element name="PayAmount" type="rac:Amount" minOccurs="1" maxOccurs="1"/>
<xs:element name="ReceivingApproved" type="rac:AttestType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="StackedInvoice" type="rac:InvoiceType" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
```

that the deprecating user is registered and the deprecation time.

##### 11.14.6. The ReceivingApproved information

Is used by the paying party for confirmation of delivery.

The [ReceivingApproved](#) contains the receiving officials name with the date of action as the XML attribute.

### 11.14.7. The `StackedInvoice` information

The `StackedInvoice` is used mainly in reimbursements, clearing accounts, credit card bills, the invoice is composed of a series of debits, with receipts and invoices collected to one invoice demanding pay. Technically these are suggested to be inserted into the main invoice.

### 11.15. The `TradeItemDataType`

The `TradeItemDataType` describes products and services.

```
<xs:complexType name="TradeItemDataType">
  <xs:sequence>
    <xs:element name="Description" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="SerialId" type="xs:token" minOccurs="0" maxOccurs="1"/>
    <xs:element name="GlobalProductId" type="xs:token" minOccurs="0" maxOccurs="1"/>
    <xs:element name="ProducersProductId" type="xs:token" minOccurs="0" maxOccurs="1"/>
    <xs:element name="SuppliersProductId" type="xs:token" minOccurs="0" maxOccurs="1"/>
    <xs:element name="DomesticProductId" type="xs:token" minOccurs="0" maxOccurs="1"/>
    <xs:element name="RevisionId" type="xs:token" minOccurs="0" maxOccurs="1"/>
    <xs:element name="CustomsId" type="xs:token" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="TaxId" type="xs:token" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Picture" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ProductDescriptionFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ProductSpecificationFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="GuaranteeFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="RepairFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="UpgradeFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="UserManualFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ProductInformationWebPage" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="TransportPackingGroup" type="xs:token" minOccurs="0" maxOccurs="1"/>
    <xs:element name="DangerousGoodsClass" type="xs:token" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="StoringTransportMessage" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="MaxTempC" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
    <xs:element name="MinTempC" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
    <xs:element name="SaftyAndEnvironmentalMessage" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="SaftyAndEnvironmentalFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="AgriculturalFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="RecyclingCode" type="xs:token" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="DisposalMessage" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="DisposalFileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="FileUrl" type="rac:FileUrl" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

EAN/GS1-codes

Many of the information items are specified in the UN/CEFACT document formats that are also expected to be linked to by URLs.

Many of the pay components are banking language impossible for non banking people to translate. So far translations here are not complete.

### 11.16. The `PayType`

Noted by the issuing party, by issuing.

The `Paytype` is the payment instruction, can be multiple and then the paying party use one of them for the payment.

Note that the issuers pay-Id of the invoice/payment is noted with the `PaymentMeans` in the `CITradeSettlementPaymentMeansType` Id field.

The content are:

```
<xs:complexType name="PayType">
  <xs:sequence>
    <xs:element name="RegDate" type="xs:date" minOccurs="0" maxOccurs="1"/>
    <xs:element name="PayDate" type="xs:date" minOccurs="1" maxOccurs="1"/>
    <xs:element name="Amount" type="rac:Amount" minOccurs="1" maxOccurs="1"/>
    <xs:element name="PaymentMeans" type="rac:CITradeSettlementPaymentMeansType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ReferenceBankAccount" type="rac:InvoiceEntryType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="ReferenceAccounting" type="rac:SeriesIdType" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
```

### 11.16.1. The **PaymentMeans** information

The **PaymentMeans** is of **CITradeSettlementPaymentMeansType** UN/CEFACT D16A Banking Payment means types and the international banking payment organisation SWIFT supplies the methods using it.

### 11.16.2. The **ReferenceBankAccount** information

Noted by the paying party, in back office.

This to allow the computer system to point at the bank account entry.

### 11.16.3. The **ReferenceAccounting** information

Used by the administrative computer systems.

This to allow the computer system to point at the accounting voucher.

## 11.17. The **DepreciationsType**

The **DepreciationsType** is used to support automatic depreciation or revaluation processes by computer programs.

The content are:

```
<xs:complexType name="DepreciationsType">
  <xs:sequence>
    <xs:element name="Deprecated" type="xs:date" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Description" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="DepreciationSchedule" type="rac:DepreciationScheduleType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="DepreciationPeriods" type="rac:DepreciationPeriodsType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Periods" type="xs:nonNegativeInteger" minOccurs="0" maxOccurs="1"/>
    <xs:element name="FirstDepreciation" type="xs:date" minOccurs="0" maxOccurs="1"/>
    <xs:element name="LastDepreciation" type="xs:date" minOccurs="0" maxOccurs="1"/>
    <xs:element name="InterestRate" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
    <xs:element name="DepreciationQuantityUnit" type="rac:TextType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Base" type="rac:Amount" minOccurs="0" maxOccurs="1"/>
    <xs:element name="BaseQuantity" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Residual" type="rac:Amount" minOccurs="0" maxOccurs="1"/>
    <xs:element name="ResidualQuantity" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Depreciations" type="rac:Amount" minOccurs="0" maxOccurs="1"/>
    <xs:element name="DepreciationsQuantity" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Balance" type="rac:Amount" minOccurs="1" maxOccurs="1"/>
    <xs:element name="BalanceQuantity" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Depreciation" type="rac:DepreciationEntryType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Revaluation" type="rac:DepreciationEntryType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="DepreciationEntryType">
  <xs:sequence>
    <xs:element name="Date" type="xs:date" minOccurs="1" maxOccurs="1"/>
    <xs:element name="Amount" type="rac:Amount" minOccurs="1" maxOccurs="1"/>
    <xs:element name="AmountQuantity" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
    <xs:element name="ReferenceBankAccount" type="rac:InvoiceEntryType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="ReferenceAccounting" type="rac:VoucherType" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
```

### 11.17.1. The **Deprecated** information

Means the depreciation is finalised..

The **OperatorInfoType** contains the accountants name with the date of action as the XML attribute.

### 11.18. The `TextType`

The `TextType` is the general tool of Rosetta multilingual accounting. One text of each language is normally allowed.

```
<xs:complexType name="TextType">
  <xs:simpleContent>
    <xs:extension base="xs:token">
      <xs:attribute name="lang" type="xs:language"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

## 12. The Alphabet Rosetta Accounting file I/O open library

Alphabet AB has decided to release the C-source code for file I/O of the Alphabet Rosetta accounting file format. It includes a `main()` and can be compiled in any operating system. It is able to read and write Rosetta, SIE and XBRL/GL files, including conversion, as demo.

The Alphabet Rosetta Accounting file I/O open library is designed as a linked list system with all data read into the main computer memory. Today memory is very large in all computers even mobile phones that the entire accounting for quite large companies can be held in the main computer memory without any technical problems. The advantage is that accounting can be handled as a word processor document rather than a data base system. To view accounting information the word processor approach is much more efficient.

However it is quite possible to make SQL templates for data base implementation, and we are willing to announce such as our solution for data base users is we get a good suggestion. As is we rather make more OS-dependent shells for other operating systems.

The Rosetta accounting file I/O open library is used together with the Alphabet Rosetta Viewer library and the Alphabet Rosetta OS-dependent shells forming the Rosetta SIE-XBRL/GL Viewer in various OS system environments and distributed as commercial products.

The basic idea of releasing an open file I/O open library for the file format (reading and writing the XML-files based on the Rosetta.xsd) is to minimise the issue of interpretation dialects of this specification. The API is far less likely to have implementation dialect issues, being used. The Rosetta accounting file I/O open library can be compiled with any ANSI C-compiler for any operating system.

It will be available on Alphabet ABs homepage [www.alphabet.se](http://www.alphabet.se) under Apache licence.