



European Committee for Standardization
Comité Européen de Normalisation
Europäisches Komitee für Normung

CEN/ISSS WS/BIIGL12

Totals

GUIDELINE

Business Domain: Post award procurement

Business Process: All

Document Identification: CEN/ISSS WS/Guideline 12

Version: 1.0

Release: 2009-11-05

Date of approval: 2009-11-05

TABLE OF CONTENTS

| | | |
|-------|---|----|
| 1 | Introduction | 3 |
| 1.1 | Purpose | 3 |
| 2 | Relevant UBL Classes and Elements | 4 |
| 2.1 | The MonetaryTotal class | 4 |
| 3 | Description | 5 |
| 3.1 | Amount totals format | 5 |
| 3.2 | Calculation Relevance | 5 |
| 3.3 | Calculation Sequence | 5 |
| 3.4 | Gross and Net amounts | 5 |
| 4 | Document level amounts | 7 |
| 4.1 | LineExtensionAmount | 7 |
| 4.2 | LineExtensionAmount in Reminder | 7 |
| 4.3 | AllowanceTotalAmount | 7 |
| 4.4 | ChargeTotalAmount | 7 |
| 4.5 | TaxExclusiveAmount | 7 |
| 4.6 | PayableRoundingAmount | 8 |
| 4.7 | TaxInclusiveAmount | 8 |
| 4.8 | PrepaidAmount | 8 |
| 4.9 | PayableAmount | 8 |
| 4.10 | Example | 8 |
| 5 | Line level amounts | 9 |
| 5.1 | Line Extension Amount | 9 |
| 5.2 | Price | 9 |
| 5.3 | Order invoice matching | 10 |
| 5.3.1 | Matching on quantity | 11 |
| 5.3.2 | Amount matching | 11 |
| 5.3.3 | Price matching | 11 |

1 Introduction

The CEN/ISSS Workshop on business interoperability interfaces for public procurement in Europe (CEN/ISSS WS/BII) is established in order to

- Identify and document the required business interoperability interfaces related to pan-European electronic transactions in public procurement expressed as a set of technical specifications, developed by taking due account of current and emerging UN/CEFACT standards in order to ensure global interoperability;
- Co-ordinate and provide support to pilot projects implementing the technical specifications in order to remove technical barriers preventing interoperability.

To facilitate implementation of electronic commerce in a standardized way, thereby enabling the development of standardized software solutions as well as efficient connections between business partners without case by case specification of the data interchange, the workshop agreed to document the required business interoperability interfaces as profile descriptions. The end goal is to reduce the cost of implementing electronic commerce to a level that is economical for small and medium size companies and institutions.

The main focus of the profile description and the associated transaction data models is to address generally expressed business requirements applicable throughout the European market. Although the profile description and associated transaction data model are designed to meet generally expressed requirements, it is still the responsibility of the users to ensure that the actual business transactions exchanges meets all the legal, fiscal and commercial requirements relevant to their business.

This guideline is one of a series of documents providing detailed guidance on the purpose and use of some key elements of the transaction data models developed by CEN/ISSS WS/BII.

All cardinalities shown in this document represent elements and associations as from the full data models; see 'Profile Architecture' for further information.

1.1 Purpose

This guideline describes the calculation of various totals in documents.

2 Relevant UBL Classes and Elements

Amount totals are included in the following document types:

- Order
- Invoice
- CreditNote
- Reminder

Normally, the amount totals are defined at the document level in these documents. They reflect the total of the amounts shown at line level. The transaction lines also contain a line total which is based on a calculation of, among other things, quantity and price for the given line.

In addition, the amount totals for taxes, charges and allowances are also specified.

In the *MonetaryTotal* class, the amount totals are defined at document level. The calculation of the amount totals is based on the following classes/elements:

- AllowanceCharge (Document and line level, and line level under the Price class)
- TaxTotal (Document and line level)
- PrepaidPayment (Document level)
- LineExtensionAmount (Line level)

2.1 The MonetaryTotal class

Note that the qualified naming of the *MonetaryTotal* class indicates the document type that it is defined within:

- Order (AnticipatedMonetaryTotal)
- Invoice (LegalMonetaryTotal)
- CreditNote (LegalMonetaryTotal)
- Reminder (LegalMonetaryTotal)

All these have identical structures. In this guideline, the name *MonetaryTotal* is used as the common term for all these classes.

3 Description

The following section contains descriptions of the use of classes and elements for amount totals.

3.1 Amount totals format

An amount total can be specified with floating decimals up to two, using the period/full stop (.) as its decimal separator.

Table 1 shows some examples:

| Amount: | format: |
|----------|----------|
| 10 | 10.00 |
| 9.465 | 9.47 |
| 9.464 | 9.46 |
| 0 | 0.00 |
| 10000.25 | 10000.25 |

The currency code must always be specified together with any amount total. The *currencyID* attribute defines the currency code according to the relevant code list. In the following example the currency code is "EUR" (Euro).

```
<cbc:PayableAmount currencyID="EUR">720.00</cbc:PayableAmount>
```

3.2 Calculation Relevance

The amount total can sometimes be calculated as the sum of all subtotals. However, some subtotals may be merely descriptive. Their purpose is to show how another subtotal is calculated. As such, these descriptive subtotals have no relevance to the total and cannot be included in the calculation of the amount total.

For example, the specification of charges and allowances at line level within the Price class may be used to permanently specify a charge for a given item. However, such values are only informative, as the charge should already be included in the calculated PriceAmount for the item in question. This means the amount is also included in the LineExtensionAmount for this line and should not be included when the amount totals are calculated in MonetaryTotal.

3.3 Calculation Sequence

Formally defining the total calculation method may be necessary if a document contains a series of interconnected taxes, allowances, or charges. For example, if an allowance is to be deducted before a charge is calculated.

To do this, the field *SequenceNumeric*, in the different class instances, is assigned a serial number beginning with 1. This indicates that the calculation of the individual taxes, charges, and allowances is performed in this numeric order.

Please note that the above mentioned calculation sequence does not influence the calculation of taxes. The tax liability of charges and allowances is always specified in the *TaxCategory*, whilst the calculation for taxes is defined by the *TaxScheme* class.

For further information on taxes, allowances, and charges, see the relevant guidelines: BII Guideline Tax and BII Guideline AllowanceCharge

3.4 Gross and Net amounts

In BII documents, prices (*PriceAmount*) are always specified as the net amount. The tax is calculated separately, and is added to the overall total (*PayableAmount*) as described below.

In the following description it is thus presumed that *LineExtensionAmount* for a given line is always exclusive of taxes.

4 Document level amounts

4.1 LineExtensionAmount

It represents the total amount for the transaction line of the document, and is calculated as the sum of the line subtotals. This can be expressed as the following calculation.

$\text{LineExtensionAmount} = \sum \text{LineExtensionAmount}$ (at line level)

The following rules apply to the line total:

- The amount is exclusive of taxes.
- The amount represents the value before adding or subtracting any relevant taxes, allowances, and charges at document level.
- The amount represent the value after adding or subtracting relevant allowances and charges at line level (as these have already been included in the individual line totals).

4.2 LineExtensionAmount in Reminder

In a Reminder document, it is possible to specify both a line amount to be credited as well as a line amount to be debited. This net amount must be considered when calculating the *LineExtensionAmount* in a Reminder, as shown by the following calculation.

$\text{LineExtensionAmount} = \sum \text{DebitLineAmount} - \sum \text{CreditLineAmount}$

The other rules for calculation of the line total are identical to those described above.

4.3 AllowanceTotalAmount

AllowanceTotalAmount defines the total of all allowances on document level. It is calculated from the calculation shown in following formula.

$\text{AllowanceTotalAmount} = \sum \text{Amount}$ (from AllowanceCharge classes on document level where ChargeIndicator = "false")

The following rules apply to the *AllowanceTotalAmount* of the document:

- The amount must not be negative

4.4 ChargeTotalAmount

ChargeTotalAmount defines the total of the charges on document level. The calculation shown is as following.

$\text{ChargeTotalAmount} = \sum \text{Amount}$ (from AllowanceCharge classes on document level where ChargeIndicator = "true")

The following rules apply to the *ChargeTotalAmount* of a document:

- The amount must not be negative

4.5 TaxExclusiveAmount

The *TaxExclusiveAmount* specifies the total amount of tax for a document. It is always specified in the field *TaxAmount* in the *TaxTotal* class on document level using the calculation described the following formula.

$\text{TaxExclusiveAmount} =$
LineExtensionAmount
- AllowanceTotalAmount
+ ChargeTotalAmount

The following rules apply to the total tax amount of the document:

- If taxes are specified in a different currency than the document currency, special rules apply. These are described in the BII Guideline Exchange Rates and Currency.

4.6 PayableRoundingAmount

PayableRoundingAmount defines any possible rounding amount that is applied to the *TaxInclusiveAmount*. The amount can be any positive or negative amount, depending on the rounding rule applied.

4.7 TaxInclusiveAmount

The *TaxInclusiveAmount* represents the total value of the purchase. The calculation is given in following formula.

TaxInclusiveAmount =

TaxExclusiveAmount
+ \sum TaxTotalAmount (document taxes that are not include in the price)
+ PayableRoundingAmount

4.8 PrepaidAmount

PrepaidAmount defines any prepaid amount applicable to the transaction. It can be represented by the following calculation.

PrepaidAmount = \sum PaidAmount (from PrepaidPayment classes on document level)

The following rules apply to the *PrepaidAmount* of a document:

- The amount must not be negative.
- When invoice are issued as paid, when paid at counter or in Internet purchase, the full amount of the invoice is considered prepaid.

4.9 PayableAmount

PayableAmount represents the total payable amount for a document. If no prepayments have been made this is the same amount as the *TaxInclusive* amount.

PayableAmount =

TaxInclusiveAmount (from the LegalTotal class on document level)
- PrepaidAmount (from the LegalTotal class on document level) +

The following rules apply to the *PayableAmount* of a document:

- Special rules apply to the specification of alternative currency codes for the *PayableAmount*. This is described in the BII Guideline Exchange Rates and Currency.
- If invoice are issued as paid, e.g. when paid at purchase with cash or payment cards, the full amount of the invoice is considered to be paid and the *Payable* amount is zero.

4.10 Example

Below is an example showing the use the *MonetaryTotal* class.

| Variable in calculation | | Sample amounts | |
|--|---|----------------|-----------------------|
| Sum of line amounts | + | 321,82 | LineExtentionamount |
| Allowance/discounts on document level | - | 9,20 | Allowances(Total) |
| Charges on document level | + | <u>7,60</u> | Charges(Total) |
| Invoice total w/o taxes excluded from unit price | = | 320,22 | TaxExclusiveAmount |
| Taxes excluded from unit price (mainly VAT) | + | 40,25 | TotalTaxes |
| Rounding of Invoice total | + | <u>-0,47</u> | PayableRoundingAmount |
| Invoice total (value of purchase) | = | 360,00 | TaxInclusiveAmount |
| Paid amounts | - | <u>120,00</u> | PrepaidAmount |
| Amount due for payment | = | 240,00 | Payable amount |

5 Line level amounts

5.1 Line Extension Amount

The line extension amount is the total amount of the line. The amount includes all allowances and charges on line level as well as taxes, other than those that are added on document level such as VAT.

The calculation for the line extension amount is the following.

LineExtentionAmount =

PriceAmount/Price_BaseQuantity x Invoiced_Quantity
+ AllowanceCharge.Amount on line where charge indicator = "true"
- AllowanceCharge.Amount one line where charge indicator = "false"

It is important to note that only the AllowanceCharge amount on the line should be added, not those that may be in the price class since those are informative, i.e. include in the price.

5.2 Price

Price. Price Amount.Amount The price amount. The price is given as net including all markups and discounts that apply to the price as well as taxes and duties other than those that are added on document level, such as VAT. The price unit is given as the unit for the Price Base Quantity.

Price. Base_Quantity.Quantity The quantity and unit to which the price applies. As example the price may be given for each kilo, in which case the unit of measure for the BaseQuantity is "KGM" and the BaseQuantity is 1. Alternatively the price could be given for each 1000 kilos in which case the BaseQuantity is 1000 "KGM".

Price. Orderable Unit Factor. Rate The factor by which the base price unit can be converted to the ordered unit. The OrderableUnitFactorRate is used to express the relationship between the invoiced quantity unit and the ordered quantity unit. If the order is made in the same unit as the invoice the OrderableUnitFactorRate is 1. If, for example, the order is made in boxes containing 24 pieces but the invoice is in pieces then the OrderableUnitFactorRate is 24.

IMPORTANT: the Unit on which the price is based must be the same as the unit for the invoice quantity.

The price unit may be different from the ordered unit in which case the Orderable Unit Factor rate is used to show the relation between the units to facilitate order-invoice matching on quantity.

When linking prices and quantities, it is fundamental to understand the Price information item and more precisely the role of the Base Quantity and Orderable Unit Factor.

$PriceAmount / Price BaseQuantity * OrderableUnitFactorRate =$ the price of one ordered unit.

When no value is assigned to BaseQuantity and OrderableUnitFactorRate it is assumed that their value is "1".

To illustrate the usage of these fields two examples are provided below in the context of Invoicing. These examples are fictional and build for the purposes of illustration. The first example aims at a rather simple way of translating ordered quantities to invoiced quantities, and the second one a more complex example.

1st example: No Conversion rate is required

This is a very straightforward simplified example. The customer orders 10 PCs. Each PC is regarded as a separate item. The supplier's base quantity (for charging) is also per item. Charges, allowances and tax information are not provided. Since no value is assigned to BaseQuantity and OrderableUnitFactorRate it is assumed that their value is "1".

```
<cac:InvoiceLine>  
  <cbc:ID>1</cbc:ID>  
  <cbc:InvoicedQuantity unitCode="EA">10</cbc:InvoicedQuantity>
```

```
<cbc:LineExtensionAmount currencyID="EUR">1450
</cbc:LineExtensionAmount>
<cac:Item>
  <cbc:Name>Desk PC</cbc:Name>
</cac:Item>
<cac:Price>
  <cbc:PriceAmount currencyID="EUR">145
  </cbc:PriceAmount>
</cac:Price>
</cac:InvoiceLine>
```

2nd example: The Conversion rate is required

This is a complex example to illustrate the usefulness of the OrderableUnitFactorRate information element. The customer orders ten barrels of oil (each containing 750 litres). Therefore for the customer the orderable unit is barrels. However, the supplier's base quantity (for charging) is not barrels but litres. Charges, allowances and tax information are not provided.

```
<cac:InvoiceLine>
  <cbc:ID>1</cbc:ID>
  <cbc:InvoicedQuantity unitCode="LTR">7500
  </cbc:InvoicedQuantity>
  <cbc:LineExtensionAmount currencyID="EUR">1125
  </cbc:LineExtensionAmount>
  <cac:Item>
    <cbc:Name>Oil Barrel </cbc:Name>
  </cac:Item>
  <cac:Price>
    <cbc:PriceAmount currencyID="EUR">150
    </cbc:PriceAmount>
    <cbc:BaseQuantity unitCode="LTR">1000
    </cbc:BaseQuantity>
  <cbc:OrderableUnitFactorRate>750</cbc:OrderableUnitFactorRate>
</cac:Price>
</cac:InvoiceLine>
```

Step by step description of the scenario:

- The quantity unit code of "LTR" in InvoicedQuantity specifies that the invoiced quantity is 7500 litres of oil.
- However, the supplier's base price (PriceAmount) is 150 euros for 1000 litres (the BaseQuantity).
- Since the buyer ordered the oil in barrels of 750 litres and not in litres, the supplier specifies in the invoice the conversion factor (OrderableUnitFactorRate) that should be applied to convert the supplier's invoiced quantity to the ordered i.e. 750. In this case, the OrderableUnitFactorRate is 750.
- Therefore the price of one barrel is $750 \text{ (OrderableUnitFactorRate)} * 150 \text{ (PriceAmount)} / 1000 \text{ (Price Base Quantity)} = 112.5$ euros per barrel (750 litres).
- The customer can match the ordered quantity to the invoiced quantity by multiplying the ordered quantity with the OrderableUnitFactorRate. i.e. $10 * 750 = 10$.
- For 10 barrels, and assuming no charges or allowances, the LineExtensionAmount would be 1125.

5.3 Order invoice matching.

Invoice can be matched to orders in the following ways.

- Matching of ordered quantity to invoiced quantity.
- Matching of anticipated monetary total to invoiced legal monetary total.
- Price matching

5.3.1 Matching on quantity

The ordered unit may differ from the invoice unit. Consequently it may be necessary to convert the quantity in the invoice to the same quantity relevant to the ordered unit. As an example one may order 10 cases of soda drinks but the suppliers invoices them by the bottle. Each case contains 12 bottles. In this case the

Order quantity = 10 CS

Invoice Quantity = 120 PCE.

Invoice.Price.Orderable unit factor rate = 12

The orderable unit factor rate element in the order is not used since it is the supplier that provides the information about this factor rate and does so in the invoice.

Consequently the calculation for the order-invoice matching is the following:

$$\text{OrderedQuantity} \times \text{OrderableUnitFactorRate} = \text{InvoicesQuantity}$$

or

$$10 * 12 = 120$$

5.3.2 Amount matching

The different totals within the anticipated monetary total of the order can be matched to the same total in legal monetary total of the invoice. As example

Matching the total amount including all charges and taxes

Order. Monetary Total. Tax Inclusive Amount = Invoice. Monetary Total. Tax Inclusive Amount

If the buyer does not know what VAT is applicable to individual items it may be more practical for him to match the total amounts excluding tax

Order. Monetary Total. Tax exclusive Amount = Invoice. Monetary Total. Tax exclusive Amount

And so forth.

5.3.3 Price matching

To be able to match the unit price for individual items the price unit in the order must be the same as the price unit in the invoice.