

Telema eDoc implementation guidelines

MANUAL FOR ERP SOFTWARE DEVELOPERS

Telema AS

15.10.2015

Version 2.08



European Union
Regional Development Fund



Investing in your future

Table of Contents

1. Introduction	3
1.1. Overview of outline and contents of this document	3
1.2. Telema EDI Module concept and assumptions	3
1.3. Telema API	3
1.4. Reading conventions and references	4
1.5. General overview of Telema eDoc	4
1.5.1. Encoding	5
1.5.2. Other notes	5
1.6. Conventions and understanding of documents per business perspectives	5
1.7. General considerations on e-document content constraints	6
2. Telema eDoc handling	7
2.1. Header - parties (sender, receiver) identification rules and conventions	7
2.2. DocumentParties	8
2.3. DocumentInfo	10
2.4. DocumentSumGroup	12
2.5. ItemEntry	13
2.5.1. ItemReserve	14
2.5.2. Item identification	15
2.5.3. Item units of measure	15
2.6. PaymentInfo	17

1. Introduction

The objective of this document is to give instructions, directions and recommendations to help in development of EDI documents in Telema eDoc – the e-document standard (or format) used when exchanging EDI messages with Telema EDI system. The document focuses only to transaction e-documents like order, despatch advice, receiving advice and invoice.

These guidelines are primary intended for analysts, designers and developers of Enterprise Resource Planning (ERP) systems.

1.1. Overview of outline and contents of this document

Chapter 1 introduces general subjects for Telema eDoc developer. Chapter 2 describes details of handing e-documents according to Telema eDoc specification. When developing Telema EDI Module for a single perspective it is good to examine also description of the opposite perspective. Details and considerations from opposite perspective can give some more understanding (or hints) as well. When general understanding of both perspectives achieved, developer can think also in terms of “direction” – sending or receiving of e-documents.

1.2. Telema EDI Module concept and assumptions

Telema EDI Module is the combination of the document conversion module converting documents between ERP System internal format and Telema eDoc format and the part of the ERP System handling the communication to the Telema EDI System.

Details about Telema EDI Module are described in *“Telema EDI Module Functional Requirements.pdf”*

1.3. Telema API

Telema API (having full name “Telema EDI SOAP Client API for .NET + COM Interop”) is developed as comprehensive yet compact and easy to use .NET class library. Telema API can be efficiently integrated in Windows platform developments in order to provide e-document exchange directly with Telema EDI System via Telema SOAP webservice. Assembly is specially designed in respect to COM registration so that it can be efficiently used not only in .NET environment (as regular class library) but also by any COM aware clients (environments). The API performs also basic validation (against eDoc schema) of e-documents.

There is developer’s reference documentation available for the API.

“TelemaClientAPI_NET_COM_DevRef_rev1-4.pdf”

1.4. Reading conventions and references

Term/Abbr.	Definition, references
EDI	Electronic Data Interchange, transfer of structured data by agreed message standards, from one computer system to another without human intervention
Telema eDoc	Electronic document specification in general described in 1.5 (and thereafter being the main subject of this document). In these guidelines the term "eDoc" is primary assumed as the specification
e-document	In general to refer to business transaction document instance represented in form of structured electronic data. In context of these guidelines, e-document is primary referred as instance of electronic document formatted in accordance to Telema eDoc; also e-order and e-invoice are referred as instances of order and invoice electronic documents according to Telema eDoc specification
Telema EDI System	In this document to be referred as system (or complex) of interconnected software and hardware resources to provide EDI service at Telema side (Telema EDI system)
EPR system	Enterprise Resource Planning, business management software. Assumed here are not only "industry standard" or well-known software packages, but any custom software system used for sales logistics business data management and automation
API	Application Programming Interface, specification intended to be used as an interface by software components to communicate with each other. An API is usually related to a software library: the API describes and prescribes the expected behavior while the library is an actual implementation of this set of rules
Telema API	In this document to refer to the Telema Client API (ref. in 1.3)
XML node	In general refers to XML node, assuming there are element nodes, attribute nodes, "text" nodes and others in XML
GLN	Global Location Number (GLN) is part of the GS1 systems of standards. It is intended to uniquely identify a location worldwide
GTIN	Global Trade Item Number (GTIN) is an identifier for trade items developed by GS1 (comprising among others of the former EAN International and Uniform Code Council)

Contents of columns in tables (green captions) describing eDoc XML nodes:

Column caption	Content
eDoc node	Telema eDoc XML node described, XPath notation used
Type	Type name of the node according to eDoc schema definition (XSD). All types found within the tables are referenced in Error! Reference source not found.
MCO	M – Mandatory node; this is not always in context of eDoc schema specification, but basically in context of the specific e-document type (or logistics character in business transaction); C – Conditional node, e.g. one/some option of nodes is required or node required in case of specific business rule; O – Optional node in context of the specific e-document (also according to eDoc schema def.)

1.5. General overview of Telema eDoc

Telema eDoc is electronic document (e-document) format for transmission within EDI messages, which relies on XML format as container for structured data. The Telema eDoc format has been derived from EDIFACT, Estonian e-invoice standard, ISO20022, and other relevant formats. It specifies different kinds of trade document types and encompasses the needs of retail companies in the Baltic States and the neighboring countries.

This document describes the essential, relevant parts of Telema eDoc specification used in most of business cases (and in respect to the 2 main e-document types). However, based on business needs, business parties (as agreed between parties) can use all the other features (data space) specified by eDoc.

Full, up to date specification of Telema eDoc is available at:

<http://www.telema.com/support/for-developer> > [Telema eDoc format description](#) (Excel file).

Please consult also examples provided by Telema technical contact.

1.5.1. Encoding

In general, the only character encoding supported for eDoc document data within Telema EDI System is **UTF-8**.

When sending e-documents, Telema API always converts encoding to UTF-8 (Telema API accepts e-documents in other encodings). When receiving e-documents from Telema EDI System (in UTF-8 encoding), Telema API can return e-documents to caller in other valid encoding (as requested by caller). Telema API documentation to be inspected for details (ref. in **1.3**).

1.5.2. Other notes

There is “Telema eDoc validation tool” available online at:

<https://test-service.telema.com/uh/documentValidator.do>

Validation solution performs document validation not only against eDoc schema definition, but also validates various moments of business logic within document data.

For validation select

- “general_invoice” for invoice validation
- “general_order” for order validation
- “general_desadv” for despatch advice validation
- “general_recadv” for receiving advice validation

Note: remove BOM (byte order mark) bytes (if present) from UTF-8 encoded eDoc file when uploading file to validator.

1.6. Conventions and understanding of documents per business perspectives

When order or invoice document is handled by a single business party, more specific type of the document is treated as per business transaction perspective of that party. Buyer from its perspective issues a “purchase order” (i.e. buyer initiates purchase transaction), but from seller’s perspective, the received order is treated as “sales order” (i.e. seller initiates sales transaction based on the order). The same also for invoice: seller issues “sales invoice” and, as per buyer perspective. Buyer receives the invoice as “purchase invoice”. Also in ERP systems document types are usually distributed as per perspective of specific business transaction.

For EDI operator (i.e. Telema), being mediator for e-document exchange between trade partners, e-document “typing” in respect to business perspective is not quite applicable. EDI operator does not change type or “character” of document during reception and forwarding. The same “invoice” delivered via EDI operator from sender’s (seller’s) perspective is “sales invoice” but from receiver’s (buyer’s) perspective is “purchase invoice”. Therefore in Telema eDoc specification for order and invoice e-documents there are only 2 general e-document types – “order” and “invoice” respectively.

Telema eDoc “order” type covers “regular order” and “return order”. Document sub-type “return order” is used for goods returning process. Returning process is initiated by the retailer, who wants to

return previously accepted goods to the supplier. “Return order” in technical perspective is recognizable by xml tag `<DocumentSubType>return</DocumentSubType>` and negative item quantities (AmountInvoiced).

Telema eDoc “desadv” type indicates to despatch advice document. Despatch advice includes information by seller to the buyer about shipped goods. This document is mostly used in 4doc supply process. EDI despatch advice must be sent to buyer before goods arrive to delivery place, where the document may be needed for acceptance process.

Telema eDoc “recadv” type indicates to receiving advice document. Receiving advice includes information by buyer to the seller about accepted goods. The document is mostly used in 4doc supply process. Receiving advice with accepted item amounts is issued by the buyer after goods acceptance process. Receiving advice could be used to activate automatic invoice creation in seller ERP system. **NB!** Accepted amounts can be different from despatched amounts (see ItemEntry/AmountAccepted data field).

Telema eDoc “invoice” type covers both “debit invoice” and “credit invoice” (also named as “credit note”). As to “credit invoice”, usage of this document sub-type is always assumed in respect to general financial accounting principle when “seller” issues credit note with reverse (negative) amounts to reduce accounts receivable against “buyer”. “Credit invoice” as per Telema eDoc specification and business perspective is not used as document issued by “buyer” for transaction like “purchase-return”. “Credit invoice” is recognizable by xml tag `<DocumentSubType>CRE</DocumentSubType>` and negative item quantities (AmountOrdered).

1.7. General considerations on e-document content constraints

One of basic questions by developer would be *“what exactly information need to be included in e-document sent to Telema?”* Answer to be found more in the fact that it is not so much Telema that needs the business transaction related data included in the e-document but the actual receiver of the e-document. Telema’s main role is to route e-documents to proper receiver. There are various trade specifics and Telema does not set constant, definite constraints of transaction data details for each and every trade business area. Telema’s eDoc specification provides extensive space to include a lot of details, both for predefined and custom usage. Good answer to this question is that implementator should move focus more to the expected requirements. These requirements derive from companies that use respective ERP system with “Telema EDI Module” implemented and business partners of those companies. Developer can also think of e-documents as their “printed” counterparts. In respective ERP system there is already definite business logic and data structures when creating e.g. “purchase order” and producing printout of such order. If data contained by the printout satisfies all of partners that receive such printout, then the same data in form of structured data in e-order should also satisfy all of respective receivers. However, some more requirements and considerations could come for identification of specific master data (e.g. clients and products coding, units of measure).

☞ **When implementing Telema EDI Module, please have in mind the specifics of your clients and their business partners, e.g. do they need to send/receive data like “Best Before”, “Lot Number”, different packaging, Deposits and/or Factoring information. Please consult Telema eDoc and personnel for additional advices.**

2. Telema eDoc handling

Main blocks of eDoc document:

```

<E-Document>
  <Header>...</Header>
  <Document>
    <DocumentType>...</DocumentType>
    <DocumentParties>
      <BuyerParty>...</BuyerParty>
      <OrderParty>...</OrderParty>
      <DeliveryParty>...</DeliveryParty>
      <SellerParty>...</SellerParty>
    </DocumentParties>
    <DocumentInfo>...</DocumentInfo>
    <DocumentSumGroup>...</DocumentSumGroup>
    <DocumentItem>
      <ItemEntry>...</ItemEntry>
      <ItemEntry>...</ItemEntry>
    </DocumentItem>
    <PaymentInfo>...</PaymentInfo>
  </Document>
</E-Document>

```

General outline (structure) of Telema eDoc document is **identical** for all transaction documents like order, despatch advice, receiving advice and invoice (and also for many other document types).

“DocumentType” element contains only text node where value indicates to document type:

- “**order**” indicates to order document;
- “**desadv**” indicates to despatch advice document;
- “**recadv**” is indicated for receiving advice document;
- “**invoice**” for invoice (also in case of credit note) document;

Only one document is allowed within single eDoc file (i.e. “E-Document” element can contain only one “Document” element).

2.1. Header - parties (sender, receiver) identification rules and conventions

Primary role of eDoc “Header” block is EDI message routing by Telema EDI System to proper receiver.

When **receiving document** you **do not** need to read data from “Header”, but instead you need to inspect appropriate party in “DocumentParties” block (see chapter 2.1): in case of received order, you need to resolve the legal party (your retail partner) from “BuyerParty” and requested delivery place from “DeliveryParty”.

However, when **sending document** you **need** to provide correct receiver ID or receiver GLN in “Header” for EDI message to be properly routed to the correct receiver. When sending document, you also need to indicate correct sender ID or sender GLN in “Header”, this is needed by Telema EDI System to ensure also that sender and receiver are linked (i.e. if e-document traffic is allowed and established for parties). When sending document, “DateIssued” must be indicated in “Header”; this date (and preferably also time part) indicates time when e-document is issued by sender system.

In general, for party identification there are two ID types available in eDoc: either GLN code ("SenderGLN", "ReceiverGLN" in "Header") or "custom code" ("SenderID", "ReceiverID" in "Header"). Whenever applicable (and available), usage of GLN codes (as globally unique codes) are preferred, however usage of "self-issued, fake GLN" codes is discouraged as this can lead to eventual duplicates (when self-assigned by different parties).

If custom codes are used by parties, these codes normally are partner internal codes from own ERP systems of parties. Parties do not have to be aware of other parties' internal codes, instead codes from both sides are communicated to Telema and Telema establishes cross-mapping of codes for the parties within Telema EDI System.

When sending document, never indicate different ID types together for sender or receiver in "Header", indicate either GLN (if available) or your internal code for a party. Always add receiver registration number and receiver country code in "Header" segment.

eDoc node	Type	MCO	Description
Header/			
DateIssued	DateTimeType	M	Date and time when e-document is issued by sender system (should include time part, mainly beneficial when tracing for troubleshooting)
SenderID	PartyReferenceType	C	Includes document sender's internal code (if GLN is not applicable), which is usually specific shop or warehouse (this code to be allocated also to "OrderParty/PartyCode" and/or "DeliveryParty/PartyCode" in document parties).
SenderGLN	GLNType	C	Includes document sender's GLN, which is usually specific shop or warehouse (this code to be allocated also to "OrderParty/GLN" and/or "DeliveryParty/GLN" in document parties).
ReceiverID	PartyReferenceType	C	Include internal code if seller's GLN code is not applicable (this code to be allocated also to "SellerParty/PartyCode" in document parties)
ReceiverGLN	GLNType	C	Include if seller's GLN code is applicable (this GLN code to be allocated also to "SellerParty/GLN" in document parties)
ReceiverRegNum	RegType	M	Receiver registration number.
ReceiverCountryCode	CountryCodeType	M	Receiver country code.

Sample Header block:

```
<Header>
  <DateIssued>2012-03-08T13:10:14</DateIssued>
  <SenderGLN>9960347466321</SenderGLN>
  <ReceiverID>332211</ReceiverID>
  <ReceiverRegNum>15123524</ReceiverRegNum>
  <ReceiverCountryCode>EE</ReceiverCountryCode>
</Header>
```

2.2. DocumentParties

Basic EDI process requires 4 parties: Buyer, Order, Delivery and Seller. Other parties can be used in more complex sales logistics processes. Consult Telema in case when designing and developing more complex logistics for Telema EDI Module. In such cases usually understanding from all involved parties needed (i.e. parties that will manage e-documents of more complex logistics).

eDoc node	MCO	Description
DocumentParties/		
OrderParty	C	Party that identifies an order sender, which is usually specific shop or warehouse of retail partner.
DeliveryParty	M	Party where the order must be delivered. Identifies and describes specific delivery place (shop) of retail partner. In most cases DeliveryParty is equal to OrderParty.
BuyerParty	M	Retail partner legal entity (and payer)
SellerParty	M	Legal entity that sells the goods ordered

Contents of party description block (most used details):

eDoc node	Type	MCO	Description
PartyCode	PartyReferenceType	C	Either PartyCode or GLN must be provided
Name	NameTextType	M	
RegNum	RegType	C	Mandatory for SellerParty, BuyerParty
VATRegNum	RegType	C	Mandatory for SellerParty, BuyerParty
GLN	GLNType	C	Either PartyCode or GLN must be provided
ContactData/			
PhoneNum	NormalTextType	O	
EmailAddress	EmailType	O	
ActualAddress/			Recommended to provide structured address data as much as applicable by particular ERP system
Address1	NormalTextLangType	M	At least basic address information must be provided for every party. In case sender's ERP System do not provide structured address info, only Address1 is used.
City	NormalTextLangType	C	
PostalCode	PostalCodeType	C	
County	NormalTextLangType	C	
CountryCode	CountryCodeType	C	
AccountInfo/			Bank account info is mandatory for SellerParty (invoice)
AccountNum	AccountType	C	Account number in local banking system
IBAN	AccountType	C	IBAN account number
BIC	BICType	C	Bank SWIFT code
BankName	NormalTextType	C	Bank name

Sample party data block:

```

<SellerParty context="self">
  <PartyCode>640</PartyCode>
  <Name>Seller Company AS</Name>
  <RegNum>10169266</RegNum>
  <VATRegNum>EE100903367</VATRegNum>
  <ContactData>
    <PhoneNum>+372 1111 111</PhoneNum>
    <EmailAddress>supplier@email.com</EmailAddress>
    <ActualAddress>
      <Address1>Saeveski 12</Address1>
      <City>Tallinn</City>
      <PostalCode>11214</PostalCode>
      <County>Harjumaa</County>
      <CountryCode>EE</CountryCode>
    </ActualAddress>
  </ContactData>
  <AccountInfo>
    <AccountNum>221001120098</AccountNum>
    <IBAN>EE332200221001120098</IBAN>
    <BIC>HABAE2X</BIC>
    <BankName>Swedbank</BankName>
  </AccountInfo>
</SellerParty>

```

Document sender must indicate “**context**” attribute for all parties. Most common types of context are “self” and “partner”. This way the attribute confirms that “**PartyCode**” IDs (values) are as from seller’s ERP system. This attribute is not relevant for document receiver.

Sample of purchase order sender context attributes for parties:

```
<BuyerParty context="self">
<OrderParty context="self">
<DeliveryParty context="self">
<SellerParty context="partner">
```

Sample of sales invoice sender context attributes for parties:

```
<BuyerParty context="partner">
<OrderParty context="partner">
<DeliveryParty context="partner">
<SellerParty context="self">
```

Most important for **document receiver** is **partner identification**. Usage of party ID type (PartyCode or GLN) can be agreed between partners (but, of course, availability of GLN codes depends if applicable in particular ERP system). Both “**PartyCode**” and “**GLN**” can be present for document party; general rule should be (assuming GLN is applicable): if “**GLN**” is provided then resolve by GLN code, otherwise by “**PartyCode**”. When resolving by “**PartyCode**”: codes in received e-order shall be the ones in your (i.e. seller’s) ERP system (this is “standard” principle when Telema EDI System performs cross-mapping of internal codes; the internal codes are preliminary communicated to Telema from both partners).

There could also be solution to identify legal party of buyer by “**RegNum**” (company registration number) of buyer. However, anyway Telema EDI System will reject incoming order document when neither “**GLN**” nor “**PartyCode**” is provided for any of the 3 mandatory parties (BuyerParty, DeliveryParty and SellerParty).

2.3. DocumentInfo

DocumentInfo block includes information about document number, relevant dates and contact info.

eDoc node	Type	MCO	Description
DocumentInfo/			
DocumentName	NormalTextType	O (all)	More detailed or custom name assigned to the document by document sender
DocumentNum	NormalTextType	M (all)	Buyer assigned reference number for the order. This number as reference to corresponding order must be later indicated in e-invoice
DocumentSubType	ShortTextType	M (invoice) C (order)	Mandatory for invoice document. Value “ DEB ” for debit invoice (value “ CRE ” for credit note). Value “ return ” refers to return order
DateInfo/		M (all)	
OrderDate	DateTimeType	M (order)	Order date
IssueDate	DateTimeType	M (all)	Date and time when e-document is issued by sender system
InvoiceDate	DateTimeType	M (invoice)	Invoice date

DueDate	DateTimeType	M (invoice)	Invoice due date
DeliveryDateRequested	DateTimeType	M (order)	Date (and possibly time) when buyer requests the goods to be delivered. Time part can also be used by buyer just to indicate some additional terms for delivery (e.g. primary/additional delivery)
DeliveryDateActual	DateTimeType	M (invoice, desadv, recadv)	Date of actual delivery of goods.
ProcessingDate	DateTimeType	M (recadv)	Document processing date
RefInfo/		C	Document reference information. Usually invoice and desadv require reference to order number.
PaymentRefNum	NormalTextType	C	Payment reference number
SourceDocument/		M (all)	
@type	DocumentTypeCode	M (all)	Reference document type. E.g. order, desadv, recadv or invoice.
SourceDocumentNum	NormalTextType	M (all)	Indicates the source document number
SourceDocumentDate	DateTimeType	O	Indicates the source document date
CreatedByContact/		M (all)	Document creator (contact person) contacts.
ContactFirstName	NameTextType	O	First name of the contact person
ContactLastName	NameTextType	O	Last name of the contact person
EmailAddress	EmailType	M (all)	Email address

Sample DocumentInfo block in invoice document:

```

<DocumentInfo>
  <DocumentName>invoice document</DocumentName>
  <DocumentNum>51500044007</DocumentNum>
  <DateInfo>
    <IssueDate>2010-04-12</IssueDate>
    <InvoiceDate>2010-04-12</InvoiceDate>
    <DueDate>2010-05-12</DueDate>
    <DeliveryDateActual>2010-04-11</DeliveryDateActual>
  </DateInfo>
  <RefInfo>
    <PaymentRefNum>112233</PaymentRefNum>
    <SourceDocument type="order">
      <SourceDocumentNum>OT123654</SourceDocumentNum>
      <SourceDocumentDate>2010-04-10</SourceDocumentDate>
    </SourceDocument>
    <SourceDocument type="desadv">
      <SourceDocumentNum>DELIV123654</SourceDocumentNum>
      <SourceDocumentDate>2010-04-11</SourceDocumentDate>
    </SourceDocument>
    <SourceDocument type="recadv">
      <SourceDocumentNum>OT123654</SourceDocumentNum>
      <SourceDocumentDate>2010-04-12</SourceDocumentDate>
    </SourceDocument>
  </RefInfo>
  <CreatedByContact>
    <ContactFirstName>John</ContactFirstName>
    <ContactLastName>Smith</ContactLastName>
    <EmailAddress>john.smith@email.com</EmailAddress>
  </CreatedByContact>
</DocumentInfo>

```

2.4. DocumentSumGroup

DocumentSumGroup is **mandatory only for invoice** document.

eDoc node	Type	MCO	Description
DocumentSumGroup/			
DocumentSum	Decimal5 FractionDigitsType	M	Total net amount. DocumentSum and sum(VAT/SumBeforeVAT) must be equal
Rounding	Decimal5 FractionDigitsType	C	Rounding for document totals
VAT/		M	Separate " VAT " block must be summarized for every VAT rate or type present within invoice lines
@vatID	VATCodeType	M	VAT type constant: TAX (Taxable), TAXEX (Tax Exemption), NOTTAX (Not Taxable)
SumBeforeVAT	Decimal5 FractionDigitsType	M	Amount base for calculating VAT
VATRate	Decimal2 FractionDigitsType	M	VAT rate
VATSum	Decimal5 FractionDigitsType	M	VAT amount
TotalVATSum	Decimal2 FractionDigitsType	M	Total VAT amount. TotalVATSum and sum(VAT/VATSum) must be equal
TotalSum	Decimal2 FractionDigitsType	M	Invoice total amount incl. VAT
Currency	CurrencyType	M	Currency code (by ISO4217)

Sample DocumentSumGroup:

```

<DocumentSumGroup>
  <DocumentSum>153.90</DocumentSum>
  <Rounding>0.00</Rounding>
  <VAT vatID="TAX">
    <SumBeforeVAT>142.40</SumBeforeVAT>
    <VATRate>20.00</VATRate>
    <VATSum>28.48</VATSum>
  </VAT>
  <VAT vatID="TAX">
    <SumBeforeVAT>9.50</SumBeforeVAT>
    <VATRate>9.00</VATRate>
    <VATSum>0.855</VATSum>
  </VAT>
  <VAT vatID="NOTTAX">
    <SumBeforeVAT>2.00</SumBeforeVAT>
    <VATRate>0.00</VATRate>
    <VATSum>0.00</VATSum>
  </VAT>
  <TotalVATSum>29.34</TotalVATSum>
  <TotalSum>183.24</TotalSum>
  <Currency>EUR</Currency>
</DocumentSumGroup>

```

It is recommended to calculate VAT sums based on neto amounts, not collect them from document rows.

2.5. ItemEntry

Within “DocumentItem” element, every “ItemEntry” describes one document line.

eDoc node	Type	MCO	Description
DocumentItem/ItemEntry/			
LineItemNum	ShortTextType	O (all)	Sequence number for document line
SellerItemCode	ShortTextType	C (all)	Product code in seller's ERP system
GTIN		C (all)	GTIN/EAN or other barcode. Usually used as primary identifier for an item.
BuyerItemCode	ShortTextType	C (all)	Product code in buyer's ERP system
ItemDescription	LongTextLangType	M (all)	Name/description of product
ItemUnitRecord/	ShortTextType	O (all)	Option to indicate more details (measures) of item and details for parent unit (e.g. “package”)
BaseUnit	ShortTextType	M (all)	Unit of measure for AmountInvoiced or AmountOrdered or AmountDespatched
AmountDespatched	Decimal5 FractionDigitsType	M (desadv)	Number of product BaseUnit's dispatched.
AmountOrdered	Decimal5 FractionDigitsType	M (order)	Number of product BaseUnit's ordered
AmountInvoiced	Decimal5 FractionDigitsType	M (invoice)	Number of product BaseUnit's invoiced. AmountInvoiced is negative for credit invoice.
AmountActual	Decimal5 FractionDigitsType	M (recadv)	The actual amount of units arrived with the delivery
AmountAccepted	Decimal5 FractionDigitsType	M (recadv)	The amount of goods accepted by receiver. AmountAccepted is base for invoicing amount in 4doc delivery process.
ItemPrice	Decimal5 FractionDigitsType	M (invoice)	Sales net price. Final price of one product or service (BaseUnit) without taxes but with discounts
ItemBasePrice	Decimal5 FractionDigitsType	C (invoice)	Price of one product or service (BaseUnit) without taxes, without discount
ItemSum	Decimal5 FractionDigitsType	M (invoice)	Total amount without taxes, with discount
Addition/		C (all)	Discount data, mandatory if present on printed invoice
@addCode	AdditionCodeType	M	Const. value “DSC”
AddContent	NormalTextType	O	Name of the discount
AddBase	Decimal5 FractionDigitsType	C	Basis of discount calculation (AddBase = ItemBasePrice * AmountInvoiced)
AddRate	Decimal2 FractionDigitsType	C	Rate of the discount (in percent)
AddSum	Decimal5 FractionDigitsType	C	Amount of the discount
VAT/		M (invoice)	VAT data
@vatID	VATCodeType	M	VAT type constant: TAX (Taxable), TAXEX (Tax Exemption), NOTTAX (Not Taxable)
SumBeforeVAT	Decimal5 FractionDigitsType	M	Amount base for calculating VAT
VATRate	Decimal2 FractionDigitsType	M	VAT rate
VATSum	Decimal5 FractionDigitsType	M	VAT amount
ItemTotal	Decimal5 FractionDigitsType	M (invoice)	Item (row) total amount (ItemSum plus taxes)

Document sender should provide all options of item identification as available from your ERP system (i.e. also BuyerItemCode if available).

See also section 2.5.2 for considerations on item identification and section 2.5.3 for considerations on item units of measure (items parsing in e-orders). Identical considerations in these sections can be applied also when e-invoice created by your Telema EDI Module is parsed by receiver (buyer) system.

2.5.1. ItemReserve

“ItemEntry/ItemReserve” is used to describe document line in more details (e.g. lot numbers, certificate data, best before/expiry dates).

One “ItemReserve” describes products with same lot number and/or best before date.

eDoc node	Type	MCO	Description
ItemEntry/ItemReserve/			
LotNum	NormalTextType	C	Lot number
ValidityDate	DateTimeType	C	Valid until date of the item
BestBefore	DateTimeType	C	Best before date of the item
BestBeforeMin	DateTimeType	C	Product best before min date. E.g. order sender may ask delivery of item with BestBefore between BestBeforeMin and BestBeforeMax
BestBeforeMax	DateTimeType	C	Product best before max date
CustomsDeclarationNum	NormalTextType	C	Customs declaration number
CertificateNum	NormalTextType	C	Certificate number
CertificateStartDate	DateType	C	Certificate valid from date
ItemReserveUnit/			
ItemUnit	ShortTextType	C	Must be the same as ItemEntry/BaseUnit
AmountActual	Decimal5 FractionDigitsType	C	Number of units with same lot number and/or best before date

Sample ItemEntry in invoice document:

```
<ItemEntry>
  <LineItemNum>1</LineItemNum>
  <SellerItemCode>11001</SellerItemCode>
  <GTIN>4740125110012</GTIN>
  <ItemDescription>Milk 2,5% 1L</ItemDescription>
  <ItemUnitRecord>
    <ItemUnit>pcs</ItemUnit>
    <UnitNumerator>1</UnitNumerator>
    <UnitDenominator>1</UnitDenominator>
  </ItemUnitRecord>
  <ItemUnitRecord>
    <!-- describes that amount in one „pack“ unit is 10/1 of „pcs“ -->
    <ItemUnit>pack</ItemUnit>
    <UnitNumerator>10</UnitNumerator>
    <UnitDenominator>1</UnitDenominator>
  </ItemUnitRecord>
  <BaseUnit>pcs</BaseUnit>
  <AmountOrdered>40.000</AmountOrdered> <!--Conditional in invoice document -->
  <AmountInvoiced>40.000</AmountInvoiced>
  <ItemPrice>0.48</ItemPrice>
  <ItemBasePrice>0.58</ItemBasePrice>
  <ItemSum>19.20</ItemSum>
  <Addition addCode="DSC">
    <AddBase>23.20</AddBase>
    <AddRate>17.24</AddRate>
    <AddSum>4.00</AddSum>
  </Addition>
  <VAT vatID="TAX">
```

```

    <SumBeforeVAT>19.20</SumBeforeVAT>
    <VATRate>20.00</VATRate>
    <VATSum>3.84</VATSum>
  </VAT>
  <ItemTotal>23.04</ItemTotal>
  <ItemReserve>
    <LotNum>04.03.2011</LotNum>
    <BestBefore>2011-03-04</BestBefore>
    <ItemReserveUnit>
      <ItemUnit>pcs</ItemUnit>
      <AmountActual>10.000</AmountActual>
    </ItemReserveUnit>
  </ItemReserve>
  <ItemReserve>
    <LotNum>06.03.2011</LotNum>
    <BestBefore>2011-03-06</BestBefore>
    <ItemReserveUnit>
      <ItemUnit>pcs</ItemUnit>
      <AmountActual>30.000</AmountActual>
    </ItemReserveUnit>
  </ItemReserve>
</ItemEntry>

```

2.5.2. Item identification

Practice shows that item (product) identification produces most of order import errors. In eDoc there are 3 primary nodes for item identification: “**SellerItemCode**”, “**GTIN**” and “**BuyerItemCode**”. To identify item, highest priority should be set to “**GTIN**”, as the barcode should be the same (and correctly entered) also in retailers ERP system. If “**GTIN**” is not available then “**SellerItemCode**” should come as next in priority to find matching item (retailers should provide code from sellers ERP system in case barcode for item is not available or not applicable). As last in identification priority come “**BuyerItemCode**”, some ERP systems (used by sellers) could have more advanced functional space to maintain item codes from ERP systems of retailers. For specific scenarios (and as agreed between partners) eDoc provides additional space (nodes) for item identification.

As agreed between partners (sender and receiver of documents) and Telema, Telema can set specific validation rules (e.g. check if required identification ID is present) for incoming documents so that invalid document lines are not received by receiving partner (such rules can result e.g. in rejection of document on invalid line, discarding of invalid lines and notifying of them by e-mail). Document line validation in Telema EDS can include not only item identification rules but also other aspects (i.e. presence and accuracy of required data entities for item as agreed between partners).

2.5.3. Item units of measure

Telema eDoc specification provides space of item structure descriptions (if needed in specific business scenarios with complex item structures (e.g. packages or even multi-level packages), and as agreed between partners), but **basic rule** is that always “**BaseUnit**” node is indicated as the unit of measure for item amount (“**AmountOrdered**”, “**AmountInvoiced**”, “**AmountDespatched**”, “**AmountAccepted**”,)

To provide information for other applicable units of measure (usually for packaging), “**ItemUnitRecord**’s” are used. “**UnitNumerator**” is used as numerator and “**UnitDenominator**” as denominator of applied fraction when calculating item amount in described unit from “base unit” (i.e. the fraction indicates amount of “base units” in described unit).

Following sample illustrates that „BaseUnit” used in order line to refer „AmountOrdered” (as well pricing) is „pkg” (assuming it is a package):

```
<ItemUnitRecord>
  <!-- amount in one „pc” unit is 1/5 of „pkg” (i.e. 5 „pc” per „pkg”) -->
  <ItemUnit>pc</ItemUnit>
  <UnitNumerator>1</UnitNumerator>
  <UnitDenominator>5</UnitDenominator>
</ItemUnitRecord>
<ItemUnitRecord>
  <!-- amount in one „kg” unit is 2/5 of „pkg” -->
  <!-- (i.e. one „pkg” is 2.5 „kg” (1 / 0.4)) -->
  <ItemUnit>kg</ItemUnit>
  <UnitNumerator>2</UnitNumerator>
  <UnitDenominator>5</UnitDenominator>
</ItemUnitRecord>
<BaseUnit>pkg</BaseUnit>
<AmountOrdered>20</AmountOrdered> <!-- 20 „pkg” ordered -->
```

To calculate amount in defined unit from indicated amount in BaseUnit, use inverted fraction; in respect to previous sample:

20 „pkg” * 5/1 = 100 „pc”
 20 „pkg” * 5/2 = 50 „kg”

Defining of unit like “kg” in this manner is shown just for better understanding that ItemUnitRecord’s with UnitNumerator and UnitDenominator can be used as universal method for units conversion, however usually this method is used only for item packaging units. For measures like item weights (as well dimensions) there is separate element block “ItemUnitRecord/UnitMeasure” (see eDoc specification), indicating item weight in this block would be more convenient in most cases.

```
<ItemUnitRecord>
  <!-- describes base unit -->
  <ItemUnit>pcs</ItemUnit>
  <UnitMeasure>
    <!-- net weight of one „pcs” is 0.45 kg -->
    <MeasureUnit>kg</MeasureUnit>
    <NetWeight>0.45</NetWeight>
  </UnitMeasure>
</ItemUnitRecord>
<ItemUnitRecord>
  <!-- 10 „pcs” in case -->
  <ItemUnit>case</ItemUnit>
  <UnitNumerator>10</UnitNumerator>
  <UnitDenominator>1</UnitDenominator>
</ItemUnitRecord>
<ItemUnitRecord>
  <!-- 1000 „pcs” on „pallet” -->
  <ItemUnit>pallet</ItemUnit>
  <UnitNumerator>1000</UnitNumerator>
  <UnitDenominator>1</UnitDenominator>
</ItemUnitRecord>
<BaseUnit>pcs</BaseUnit>
```

There are no constraints or rules set by eDoc as to **unit codes naming**, it is language (country) specific, as well in respect to conventions in partner ERP configuration. Implementing e-order data automated parsing functionality, in case of possible structured items and variety of item units, for a solid, automated EDI client solution attention should be put to capability of enumeration, identification (and possible conversion) of units in received orders.

2.6. PaymentInfo

Mandatory segment for all invoice senders in Estonia. PaymentInfo segment is used for generating payment.

eDoc node	Type	MCO	Description
PaymentInfo/			
Currency	CurrencyType	M (invoice)	Three-character currency code
PaymentRefID	RefType	C (invoice)	Payment reference number. PaymentRefID or PaymentDescription must exist.
PaymentDescription	PaymentDescriptionType	C (invoice)	Description used as payment detail
Payable	YesNoType	M (invoice)	YES- invoice needed to be paid NO – invoice does not need payment (used for information purposes)
PayDueDate	DateType	M (invoice)	Payment due date
PaymentTotalSum	Decimal2FractionDigitsType	M (invoice)	Total amount of payment
PayerName	NameTextType	M (invoice)	Name of the payer
PaymentID	NormalTextType	M (invoice)	Invoice number
PayToAccount	AccountType	M (invoice)	The beneficiary's account number
PayToName	NameTextType	M (invoice)	The beneficiary's name

```

<PaymentInfo>
  <Currency>EUR</Currency>
  <PaymentRefID>123456</PaymentRefID>
  <PaymentDescription>Payment for invoice no 51500044007</PaymentDescription>
  <Payable>YES</Payable>
  <PayDueDate>2010-05-12</PayDueDate>
  <PaymentTotalSum>183.24</PaymentTotalSum>
  <PayerName>Buyer Company AS</PayerName>
  <PaymentID>51500044007</PaymentID>
  <PayToAccount>221001120098</PayToAccount>
  <PayToName>Seller Company AS</PayToName>
</PaymentInfo>

```