

PEPPOL

PAN-EUROPEAN PUBLIC PROCUREMENT ONLINE



BIS Order only - 3A

OpenPEPPOL AISBL, Post-Award Coordinating Community
Version 3.0.0

Table of Contents

1. Introduction to openPEPPOL and BIS
 - 1.1. Audience
2. Principles and prerequisites
 - 2.1. Scope
 - 2.2. Parties and roles
 - 2.3. Benefit
 - 2.4. Interoperability
3. Process and typical use cases
 - 3.1. Process flow
 - 3.2. Business process Diagram
 - 3.3. Use case 1 – Ordering of numbered items/articles
 - 3.4. Use case 2 – Ordering of free text articles
 - 3.5. Use case 3 – Ordering of services
 - 3.6. Use case 4 – Complex ordering
4. Semantic datatypes
 - 4.1. Primitive types
 - 4.2. Semantic data types
5. Code lists
 - 5.1. Code lists for coded elements
 - 5.2. Code list for identifiers
6. Description of selected parts of the order message
 - 6.1. Parties
 - 6.2. Attachments
 - 6.3. Product identification
 - 6.4. Product name and description
 - 6.5. Quantities and units
 - 6.6. Prices
 - 6.7. Allowances and charges
 - 6.8. Calculation of totals (AnticipatedMonetaryTotals)
 - 6.9. Tax total
 - 6.10. Line VAT Category
7. Peppol Identifiers
 - 7.1. Profiles and messages
 - 7.2. Customization and Profile identifiers
 - 7.3. Namespaces

The PEPPOL Business Interoperability Specification, “BIS” from here on after, has been developed by the OpenPEPPOL AISBL Post Award Coordinating Community and is published as part of the PEPPOL specifications.

1. Introduction to openPEPPOL and BIS

This BIS is a result of work within [OpenPEPPOL](https://peppol.eu/about-openpeppol/?rel=tab41) (<https://peppol.eu/about-openpeppol/?rel=tab41>) and is published as part of the [PEPPOL](https://peppol.eu/?rel=undefined) (<https://peppol.eu/?rel=undefined>) specifications.

This PEPPOL BIS provides a set of specifications for implementing a [PEPPOL](https://peppol.eu/?rel=undefined) (<https://peppol.eu/?rel=undefined>) business process. The document is concerned with clarifying requirements for ensuring interoperability of pan-European Public eProcurement and provides guidelines for supporting these requirements and how to implement them. This PEPPOL BIS is based on the CEN WS/BII Profile “BII Profile 03 Order Only” [CEN BII2 Profile 03. Order Only](#).

The purpose of this document is to describe a common format for the order message in the European market, and to facilitate an efficient implementation and increased use of electronic collaboration regarding the ordering process based on this format.

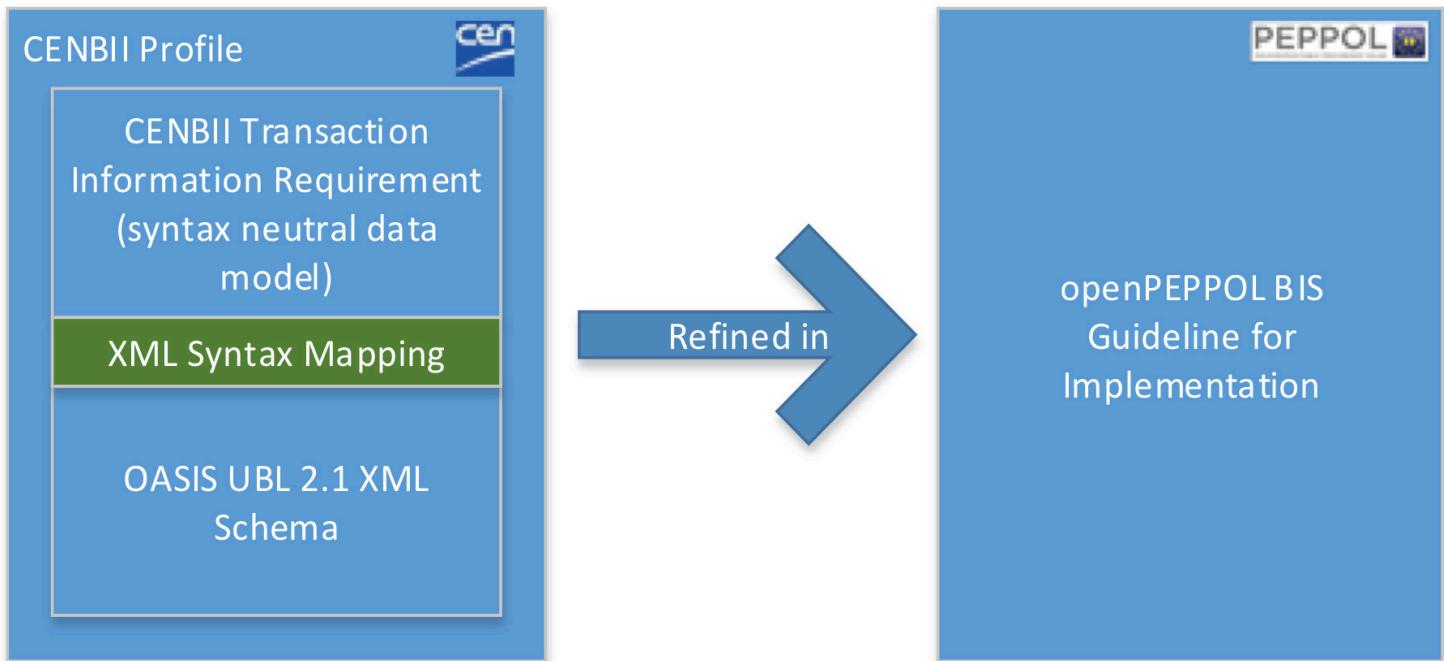


Figure 1. Relationship between BII profiles and PEPPOL BIS

1.1. Audience

The audience for this document is organizations wishing to be [PEPPOL](https://peppol.eu/?rel=undefined) (<https://peppol.eu/?rel=undefined>) enabled for exchanging electronic business documents, and/or their ICT-suppliers.

These organizations may be:

- Service providers
- Contracting Authorities
- Economic Operators
- Software Developers

More specifically it is addressed towards the following roles:

- ICT Architects
- ICT Developers

- Business Experts

For further information, please see [PEPPOL BIS common text and introduction](#)

(<https://joinup.ec.europa.eu/svn/peppol/PEPPOL%20BIS%20Common%20text%20and%20introduction%20-%20ver%201%202014-04-14.pdf>).

2. Principles and prerequisites

This chapter describes the principles and assumptions that underlie the use of PEPPOL Ordering. It is based on the [CEN BII2 Profile 03, Order Only](#).

2.1. Scope

This BIS describes a process comprising a Buyer to issue an electronic order without an order confirmation by the Seller.

The main activities supported by this profile are:

Structured Ordering

The Order transaction should support the structured ordering of goods or services, using free text or use of identifiers. The information source of the ordered products may be a (paper or electronic) catalogue.

Accounting

The ordering process must support the allocation of budgets, so the value amounts of the ordered products may be stated. The buyer may provide some information that the seller is required to place on the invoice for aiding and automation of invoice processing.

Invoice Verification

The buyer may provide some information that the seller is required to place on the invoice for aiding and automation of invoice approval.

VAT reporting

VAT reporting is not a general requirement on orders. The level of support in orders is to

- Enable VAT reporting in invoices by providing VAT number of buyer in case of reverse charges.
- VAT can be stated as an estimate to enable buyers to state expected value of order. This can be helpful in automated matching of orders and invoices. VAT information is informative and does not affect the terms of trade.

Transport and delivery

Only limited support is in scope for transport related information, but it is recognized that the buyer needs to be able to provide some information about requested delivery location, some basic term, time and contact persons for a delivery of an order.

Inventory

Supporting inventory management is not in scope, but structured orders based on catalogues can be used to automate picking at supplier warehouses.

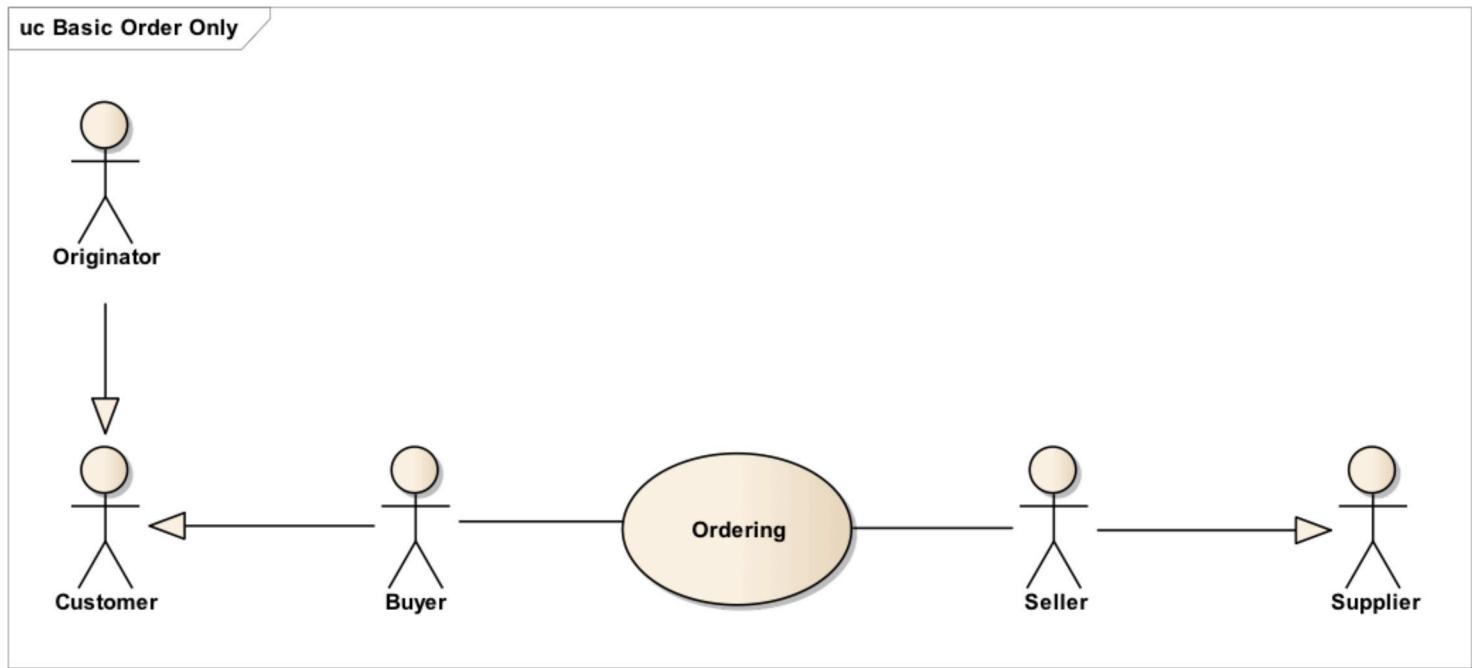
2.2. Parties and roles

The table below gives the definitions of the parties and roles of the ordering process.

Business partners	Description

Customer	The customer is the legal person or organization who is in demand of a product or service. Examples of customer roles: buyer, consignee/delivery part, debtor, contracting authority, originator.
Supplier	The supplier is the legal person or organization who provides a product or service. Examples of supplier roles: seller, consignor, creditor, economic operator.
Role/actor	Description
Buyer cac:BuyerCustomerParty	The buyer is the legal person or organization acting on behalf of the customer and who buys or purchases the goods or services.
Seller cac:SellerSupplierParty	The seller is the legal person or organization acting on behalf of the supplier and who sells goods or services to the customer.
Originator cac:OriginatorCustomerParty	A person or unit that initiates an order.
Invoicee cac:AccountingCustomerParty	A person or unit that receives the invoice (invoicee) on behalf of the customer.

The following diagram links the business processes to the roles performed by the Business Partners.



2.3. Benefit

Based on success with automation of invoicing, there is a growing interest in automation of ordering also. This approach has two dimensions: Support further automation of invoicing and using structured catalogues as basis for ordering. Implementing this BIS is an important step for many companies and government agencies towards full procurement automation.

For the sellers, the approval, picking and invoicing can be automated significantly.

For the procuring agency, approval and accounting of invoices can be automated and ordering can be structured by use of catalogues.

Other potential benefits of using this BIS are, among others:

- Can be used by procuring agencies as step towards automation of procurement. The flexibility of the specifications allows the buyers to gradually automate and structure ordering, based on a cost/benefit approach.
- SME can offer their trading partners the option of exchanging standardized documents in a uniform way and thereby move all orders into electronic form.
- Large companies can implement this BIS as standardized documents for general operations and implement custom designed bi-lateral connections for large trading partners.
- Can be used as basis for restructuring of in-house processes of orders and invoices.
- Significant saving can be realized by the procuring agency by automating and streamlining in-house processing.
- Significant saving can be realized by the sellers by automating and streamlining in-house processing. Linking to picking and invoicing can be improved significantly based on increased order quality, restructuring of invoice dispute resolution and shorter payment cycles.
- For the procuring agency, invoice automation and ordering can be structured.

2.4. Interoperability

This PEPPOL BIS structure is based on the European Interoperability Framework 2.0 [EIF](#) (<http://ec.europa.eu/idabc/en/document/2319/5644.html>). This PEPPOL BIS applies the Framework as follows:

Legal Interoperability

- Legal:
 - In implementations supporting public sector buyers, the use of the Punch out BIS has to be monitored in order to secure that the buyers act in line with EU procurement directives.

Organizational interoperability

- Organization (Organization/Business):
 - This PEPPOL BIS supports B2B and B2G.
 - This PEPPOL BIS supports cross border, regional and domestic ordering in EU and EEA.
 - This PEPPOL BIS can function as a component in an EDI agreement within a trading community.
 - This PEPPOL BIS supports linking of business processes within the sending and receiving organization. The process of order transmission in electronic form can be linked into internal processes of both sender and receiver, which may differ for various reasons.
- Organization (Process):
 - This PEPPOL BIS supports a set of “common business processes” that is assumed to be supported by most enterprises whether public or private. These are processes that are used widely or understood as being relevant for most companies.

Semantic interoperability

- Semantic: The set of information elements is assumed to be sufficient to support organizational business and processing requirements stated above.

- A CORE business cart message:
 - Data model, a set of elements that the receiver MUST be able to process.
 - Business rules, a set of business rules that ensure a common way of processing the information elements. The rules are stated in a way that allows for automated validation of document instances. Issuers and receivers can verify that the exchanged document conforms to the rules of this BIS.
PEPPOL adds business rules on top of the data model to clarify certain design choices left open by the {cenbii}. These choices are intended to lower the implementation threshold by limiting options for implementers and thereby increase interoperability of PEPPOL punch out transactions.

Technical interoperability

- Technical Interaction (Process and semantic implementation):
 - Binding to OASIS UBL 2.1, see [UBL 2.1](#) (<http://docs.oasis-open.org/ubl/UBL-2.1.html>)
 - ISO/IEC 19757-3 Schematron, for automation of document validation, see [Schematron](#) (<http://schematron.com>)
- Technical Interaction (eSignature Validation):
 - Not mandatory in this PEPPOL BIS. Not supported.

3. Process and typical use cases

The Order only profile includes the sending of Orders from a Buyer to a Seller.

3.1. Process flow

The Order process flow can be described as follows:

- A Buyer submits an Order to the Seller requesting for delivery of goods or services
- An Order may refer to a framework agreement for its terms and conditions; otherwise the Buyer's terms and conditions apply.
- An Order may contain items (goods or services) with item identifiers or items with free text description.

3.2. Business process Diagram

3.2.1. Legend for BPMN diagrams

The diagrams are expressed in the BPMN notation. The diagram below serves as an explanation for the diagrams used in the process descriptions.

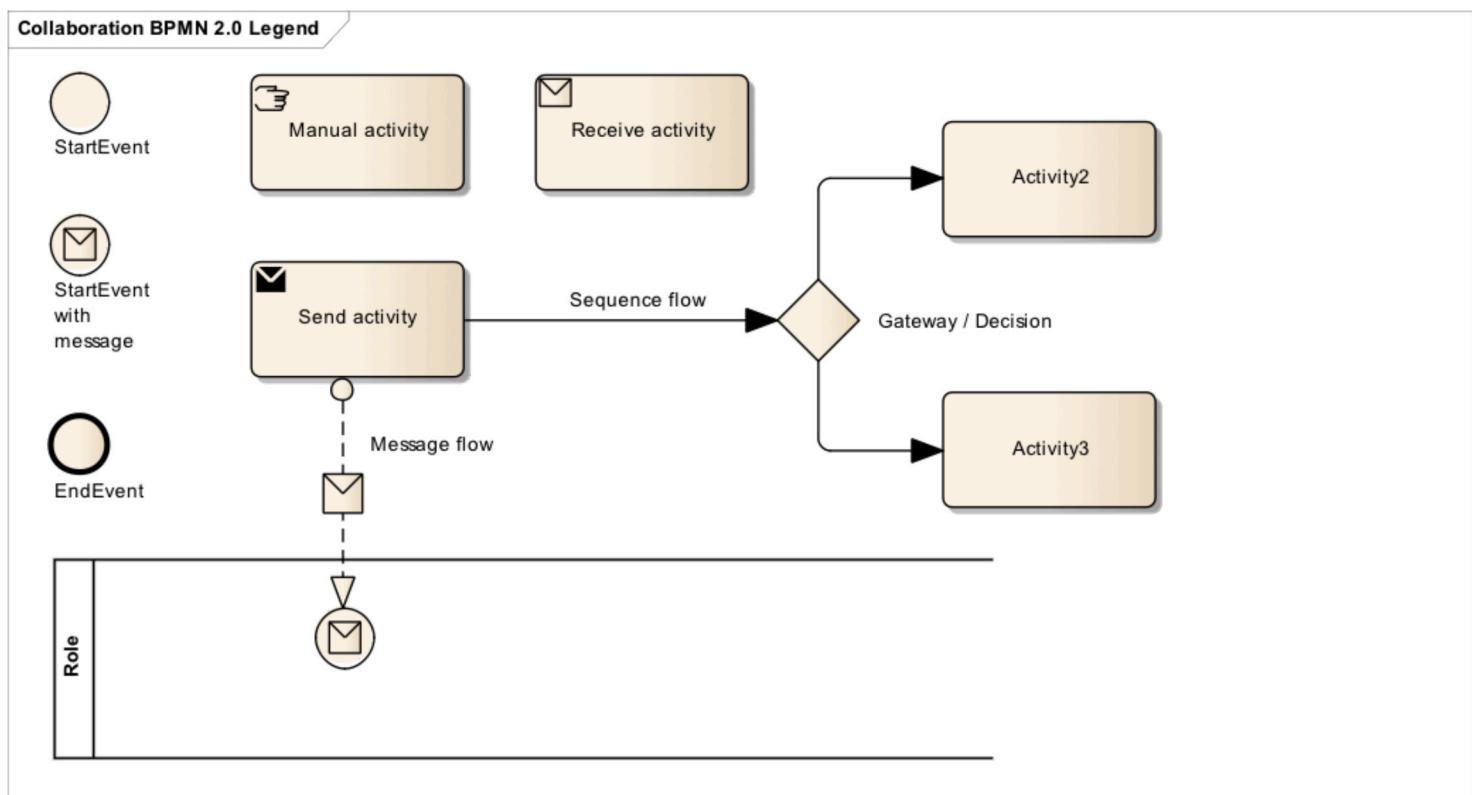
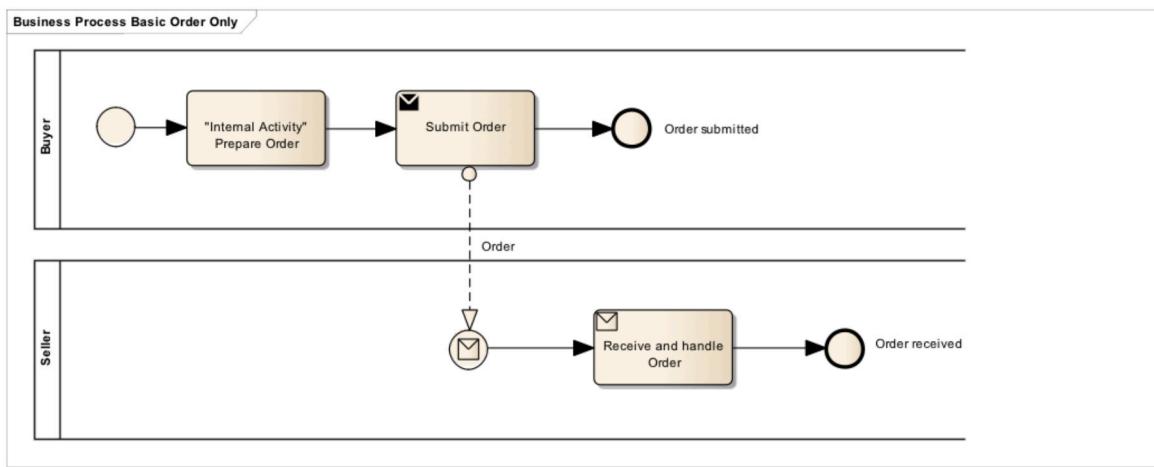


Figure 2. BPMN legend

The following diagram shows the choreography of the business process implemented by the BIS.



3.3. Use case 1 – Ordering of numbered items/articles

This use case contains an order of numbered items/articles.

Use Case number	1
Use Case Name	Ordering of numbered items/articles
Use Case Description	<ul style="list-style-type: none"> An order of numbered articles. The order instructs the seller of the delivery address. The seller can deliver some of the items but not all. One item needs to be replaced.
Parties involved	Buyer Seller
Assumptions	<p>The buyer has a catalogue or list of products to order. The catalogue contains the item numbers, names and type of unit of measure.</p>
The flow	<p>The buyer creates the order with 3 different lines and items. The seller receives the order.</p>
Result	<p>The buyer and the seller has reached an agreement. If the invoice has an order reference, the invoice can be matched automatically.</p>
XML example file	<u>Use case example files</u> (https://github.com/OpenPEPPOL/poacc-upgrade-3/tree/master/rules/use-case-examples/order)

3.4. Use case 2 – Ordering of free text articles

This use case contains an order of free text articles.

Use Case number	2
Use Case Name	Ordering of free text articles

Use Case number	2
Use Case Description	An order with item/articles described in free text and attribute/value pairs.
Parties involved	Buyer Seller Originator
Assumptions	The buyer does not have structured item information. The buyer must specify the items in a way that ensures that the seller can properly identify the requested items.
The flow	The buyer creates the order with 2 different lines and items. The seller receives the order.
Result	The buyer and the seller has reached an agreement. If the invoice has an order reference, the invoice can be matched automatically.
XML example file	<u>Use case example files</u> (https://github.com/OpenPEPPOL/poacc-upgrade-3/tree/master/rules/use-case-examples/order)

3.5. Use case 3 – Ordering of services

This use case contains an order of services.

Use Case number	3
Use Case Name	Ordering of services
Use Case Description	An order of translation services. Delivery location and period is specified.
Parties involved	Buyer Seller
Assumptions	The buyer is using a form with pre-defined and agreed properties for this service.
The flow	The buyer creates the order with one line requesting translation between Swedish and Spanish. The seller receives the order.
Result	The buyer and the seller has reached an agreement. If the invoice has an order reference, the invoice can be matched automatically.
XML example file	<u>Use case example files</u> (https://github.com/OpenPEPPOL/poacc-upgrade-3/tree/master/rules/use-case-examples/order)

3.6. Use case 4 – Complex ordering

This use case contains an order with almost all elements in the order message used.

Use Case number	4
Use Case Name	Complex ordering
Use Case Description	An order for numbered items with allowance and charges both on order level, line level and price.
Parties involved	Buyer Seller Originator
Assumptions	The buyer has a catalogue or list of products to order. The catalogue contains the item numbers, names and type of unit of measure. The buyer has reached a special agreement with the seller regarding discounts on the order, order lines and price.
The flow	The buyer creates the order with 4 different lines and items. The seller receives the order.
Result	The buyer and the seller has reached an agreement. If the invoice has an order reference, the invoice can be matched automatically.
XML example file	<u>Use case example files</u> (https://github.com/OpenPEPPOL/poacc-upgrade-3/tree/master/rules/use-case-examples/order)

4. Semantic datatypes

Semantic data types are used to bridge the gap between the semantic concepts expressed by the information elements and the technical implementation. The semantic data types define the allowed value domain for the content, and any additional information components (attributes) needed in order to ensure its precise interpretation.

4.1. Primitive types

Semantic data type content may be of the following primitive types. These primitive types were taken from [ISO 15000-5:2014](#) (<https://www.iso.org/standard/61433.html>), Annex A.

Primitive type	Definition
Binary	A set of finite-length sequences of binary digits.
Date	Time point representing a calendar day on a time scale consisting of an origin and a succession of calendar ISO 8601:2004 (https://www.iso.org/standard/40874.html).
Decimal	A subset of the real numbers, which can be represented by decimal numerals.
String	A finite sequence of characters.

4.2. Semantic data types

The different semantic data types are described in the tables below, where various features such as attributes, format, and decimals as well as the basic type are defined for each semantic data type. They are based on [ISO 15000-5:2014](#) (<https://www.iso.org/standard/61433.html>).

When used in an instance document, each data element will contain data. In the below tables this is identified as the “content”. Whenever a business term is used this term shall **always** have content and therefore the content is always mandatory.

4.2.1. Amount

An amount states a numerical monetary value. The currency of the amount is defined as a separate business term.

!
Amount is floating up to two fraction digits.

Component	Use	Primitive Type	Example
Content	Mandatory	Decimal	10000.25

4.2.2. Price Amount

A price amount states a numerical monetary amount value for data elements that contain item prices that may be multiplied by item quantities. The currency of the amount is defined as a separate business term.



Unit price amount does not set restrictions on number of decimals, as contrast to the Amount type

Component	Use	Primitive Type	Example
Content	Mandatory	Decimal	10000.1234

4.2.3. Percentage

Percentages are given as fractions of a hundred (per cent) e.g. the value 34,78 % in percentage terms is given as 34,78.



No restriction on number of decimals for percentages.

Component	Use	Primitive Type	Example
Content	Mandatory	Decimal	34.7812

4.2.4. Quantity

Quantities are used to state a number of units such as for items. The code for the Unit of Measure is defined as a separate business term.



No restriction on number of decimals for quantities.

Component	Use	Primitive Type	Example
Content	Mandatory	Decimal	10000.1234

4.2.5. Code

Codes are used to specify allowed values in elements as well as for lists of options. Code is different from Identifier in that allowed values have standardized meanings that can be known by the recipient.



Codes shall be entered exactly as shown in the selected code list

Component	Use	Primitive Type	Example
Content	Mandatory	String	Abc123

4.2.6. Identifier

Identifiers (IDs) are keys that are issued by the sender or recipient of a document or by a third party.



The use of the attributes is specified for each information element.

Component	Use	Primitive Type	Example
Content	Mandatory	String	abc:123-DEF
Scheme identifier	Conditional	String	GLN
Scheme version identifier	Conditional	String	1.0

4.2.7. Date

Dates shall be in accordance to the “Calendar date complete representation” as specified by [ISO 8601:2004](#) (<https://www.iso.org/standard/40874.html>), format **YYYY-MM-DD**.



Dates shall **not** include timezone information.

Table 1. EN 16931_ Date. Type

Component	Use	Primitive Type	Example
Content	Mandatory	Date	2017-12-01

4.2.8. Time

Time shall be in accordance to the “Extended time format” as specified by [ISO 8601:2004](#) (<https://www.iso.org/standard/40874.html>), format [hh]:[mm]:[ss].



Time shall **not** include timezone information. Decimal fraction on seconds SHALL not be used.

Table 2. EN 16931_ Date. Type

Component	Use	Primitive Type	Example
Content	Mandatory	Date	09:30:12

4.2.9. Document Reference

Document Reference Types are identifiers that were assigned to a document or document line.

Table 3. Document Reference. Type

Component	Use	Primitive Type	Example
Content	Mandatory	String	abc:123-DEF

4.2.10. Text

Text is the actual wording of anything written or printed. Line breaks in the text may be present, and any line breaks should be preserved and respected by the receiver's system

Component	Use	Primitive Type	Example
Content	Mandatory	String	5% allowance when paid within 30 days

4.2.11. Binary objects

Binary objects can be used to describe files which are transmitted together with the business document. Attachments shall be transmitted together with the business document. The binary object has two supplementary components: a Mime Code, which specifies the Mime type of the attachment and a Filename that is provided by (or on behalf of) the sender of the business document.

Component	Use	Primitive Type	Example
Content	Mandatory	Binary	QmFzZTY0IGNvbnRlbnQgZXhhbXBsZQ==
Mime Code	Mandatory	String	image/jpeg
Filename	Mandatory	String	drawing5.jpg

4.2.12. Boolean

Boolean indicators are used to specify the two allowed values, *true* or *false*. All elements of datatype Boolean, must have either *true* or *false* as their value.

Component	Use	Primitive Type	Example
Content	Mandatory	String	true

5. Code lists

5.1. Code lists for coded elements

Any element with the semantic data type = **code**, can mandate the use of a specific code list (or a fixed value). The applicable code lists can be found in the [Code list section](http://test-docs.peppol.eu/poacc/upgrade-3/codelist/) (<http://test-docs.peppol.eu/poacc/upgrade-3/codelist/>). In this section, you can find the valid codes, their names and description, and also links to where the same code list is used elsewhere in the transaction, or in other PEPPOL BIS v3. documents.

5.2. Code list for identifiers

5.2.1. Party identifiers and party legal registration identifier scheme

All party identifiers (`cac:PartyIdentification/cbc:ID`) and party legal registration identifier (`cac:PartyLegalEntity/cbc:CompanyID`) has an optional scheme identifier attribute (`@schemeID`). If used, the value shall be chosen from the code list [ICD codes](http://test-docs.peppol.eu/poacc/upgrade-3/codelist/ICD/) (<http://test-docs.peppol.eu/poacc/upgrade-3/codelist/ICD/>)

Examples of usage in `cac:PartyIdentification`

```
<cac:PartyIdentification>
    <cbc:ID schemeID="0088">5790000435968</cbc:ID> 1
</cac:PartyIdentification>
```

XML

- ¹ schemeID attribute is optional, but when used, the codes must be from [ICD codes](http://test-docs.peppol.eu/poacc/upgrade-3/codelist/ICD/) (<http://test-docs.peppol.eu/poacc/upgrade-3/codelist/ICD/>)

5.2.2. Electronic address identifier scheme identifier

All electronic address identifiers (`cbc:EndpointID/@schemeID`) use of a code list to be maintained by CEF.



This code list is still not available. Use PEPPOL Policy for identifiers until code list is available

Examples of usage in `cbc:EndpointID`

```
<cbc:EndpointID schemeID="DK:CVR">DK87654321</cbc:EndpointID> 1
```

XML

- ¹ schemeID attribute is mandatory

6. Description of selected parts of the order message

6.1. Parties

The following parties/roles may be specified in the message:

6.1.1. SellerSupplierParty (Seller)

The seller is the legal person or organization acting on behalf of the supplier and who sells goods or services to the buyer. The seller is mandatory in the PEPPOL BIS Order message.

Example of seller information

```
<cac:SellerSupplierParty>
  <cac:Party>
    <cbc:EndpointID schemeID="NO:ORGNR">222333222</cbc:EndpointID>
    <cac:PartyIdentification>
      <cbc:ID schemeID="0192">N0222333222</cbc:ID>
    </cac:PartyIdentification>
    <cac:PostalAddress>
      <cbc:StreetName>Harbour street</cbc:StreetName>
      <cbc:AdditionalStreetName>Dock 45</cbc:AdditionalStreetName>
      <cbc:CityName>Bergen</cbc:CityName>
      <cbc:PostalZone>5005</cbc:PostalZone>
      <cbc:CountrySubentity>Region West</cbc:CountrySubentity>
      <cac:AddressLine>
        <cbc:Line>Gate 34</cbc:Line>
      </cac:AddressLine>
      <cac:Country>
        <cbc:IdentificationCode>NO</cbc:IdentificationCode>
      </cac:Country>
    </cac:PostalAddress>
    <cac:PartyLegalEntity>
      <cbc:RegistrationName>Cod Liver Oil Limited</cbc:RegistrationName>
    </cac:PartyLegalEntity>
    <cac:Contact>
      <cbc:Name>Øystein</cbc:Name>
      <cbc:Telephone>+47555444333</cbc:Telephone>
      <cbc:ElectronicMail>oystein@codliveroil.no</cbc:ElectronicMail>
    </cac:Contact>
  </cac:Party>
</cac:SellerSupplierParty>
```

XML

6.1.2. BuyerCustomerParty (Buyer)

The buyer is the legal person or organization acting on behalf of the customer and who buys or purchases the goods or services. The buyer is mandatory in the PEPPOL BIS Order message.

Example of buyer information

```

<cac:BuyerCustomerParty>
  <cac:Party>
    <cbc:EndpointID schemeID="SE:ORGNR">5541277711</cbc:EndpointID>
    <cac:PartyIdentification>
      <cbc:ID schemeID="0007">5541277711</cbc:ID>
    </cac:PartyIdentification>
    <cac:PartyName>
      <cbc:Name>City Hospital</cbc:Name>
    </cac:PartyName>
    <cac:PartyLegalEntity>
      <cbc:RegistrationName>City Hospital 345433</cbc:RegistrationName>
      <cbc:CompanyID schemeID="0007">5541277711</cbc:CompanyID>
      <cac:RegistrationAddress>
        <cbc:CityName>Eurocity</cbc:CityName>
        <cac:Country>
          <cbc:IdentificationCode>SE</cbc:IdentificationCode>
        </cac:Country>
      </cac:RegistrationAddress>
    </cac:PartyLegalEntity>
    <cac:Contact>
      <cbc:Name>Martin Foggerty</cbc:Name>
      <cbc:Telephone>+46555785488</cbc:Telephone>
      <cbc:ElectronicMail>martin.foggerty@cityhospital.se</cbc:ElectronicMail>
    </cac:Contact>
  </cac:Party>
</cac:BuyerCustomerParty>

```

6.1.3. OriginatorCustomerParty (Originator)

The unit initiating the order. Most often the end user. The originator information is optional in the PEPPOL BIS Order message.

Example of originator information

```

<cac:OriginatorCustomerParty>
  <cac:Party>
    <cac:PartyIdentification>
      <cbc:ID schemeID="0088">7300010000001</cbc:ID>
    </cac:PartyIdentification>
    <cac:PartyName>
      <cbc:Name>Surgery Department</cbc:Name>
    </cac:PartyName>
    <cac:Contact>
      <cbc:Name>Dr Bengt</cbc:Name>
      <cbc:Telephone>+46555444777</cbc:Telephone>
      <cbc:ElectronicMail>bengt@cityhospital.no</cbc:ElectronicMail>
    </cac:Contact>
  </cac:Party>
</cac:OriginatorCustomerParty>

```

AccountingCustomerParty (Invoicee)

The invoicee is the legal person or organization acting on behalf of the customer and who receives the invoice for the order. The invoicee information is optional in the PEPPOL BIS Order message.

Example of invoicee information

```

<cac:AccountingCustomerParty>
  <cac:Party>
    <cbc:EndpointID schemeID="SE:ORGNR">5544332211</cbc:EndpointID>
    <cac:PartyIdentification>
      <cbc:ID schemeID="0007">5544332211</cbc:ID>
    </cac:PartyIdentification>
    <cac:PartyName>
      <cbc:Name>Swedish Hospitals</cbc:Name>
    </cac:PartyName>
    <cac:PostalAddress>
      <cbc:StreetName>High Street 23</cbc:StreetName>
      <cbc:AdditionalStreetName>First floor</cbc:AdditionalStreetName>
      <cbc:CityName>Trondheim</cbc:CityName>
      <cbc:PostalZone>7005</cbc:PostalZone>
      <cbc:CountrySubentity>Region M</cbc:CountrySubentity>
      <cac:AddressLine>
        <cbc:Line>Room 18</cbc:Line>
      </cac:AddressLine>
      <cac:Country>
        <cbc:IdentificationCode>NO</cbc:IdentificationCode>
      </cac:Country>
    </cac:PostalAddress>
    <cac:PartyLegalEntity>
      <cbc:RegistrationName>Swedish Hospitals AB</cbc:RegistrationName>
      <cbc:CompanyID>5544332211</cbc:CompanyID>
      <cac:RegistrationAddress>
        <cbc:CityName>Stockholm</cbc:CityName>
        <cac:Country>
          <cbc:IdentificationCode>SE</cbc:IdentificationCode>
        </cac:Country>
      </cac:RegistrationAddress>
    </cac:PartyLegalEntity>
  </cac:Party>
</cac:AccountingCustomerParty>

```



In order to facilitate the invoicee information to be used in the invoice it is recommended to include as much information as possible, ie. PostalAddress, PartyTaxScheme and PartyLegalEntity in addition to PartyName and PartyIdentification.

6.2. Attachments

Non-XML documents can be sent as attachments to the PEPPOL BIS Order. This could be drawings or time sheets or other documents relevant for the order. The attachment can either be sent as a binary object encoded in Base64 embedded in the message or as a URI to an external address as a link.



It is recommended to send attachments as embedded, binary objects and not as external references.

Valid codes can be found in [Code list section](http://test-docs.peppol.eu/poacc/upgrade-3/codelist/) (<http://test-docs.peppol.eu/poacc/upgrade-3/codelist/>).

Example of attachment as an embedded, binary object

```
<cac:AdditionalDocumentReference>
  <cbc:ID>100</cbc:ID>
  <cbc:DocumentType>Drawing</cbc:DocumentType> 1
  <cac:Attachment>
    <cbc:EmbeddedDocumentBinaryObject mimeCode="application/pdf" filename="blueprint.pdf"
>Ymx1ZXByaW50</cbc:EmbeddedDocumentBinaryObject> 2
  </cac:Attachment>
</cac:AdditionalDocumentReference>
```

- 1 It is recommended to use element `cac:AdditionalDocumentReference/cbc:DocumentType` to send a short description of the content of the attachment.
- 2 File name and extension should be sent in the `filename` attribute to the `cbc:EmbeddedDocumentBinaryObject` element.



Attachments should be used for additional information and not as order copies.

6.3. Product identification

Product identification must be done using the identifiers described below:

- Sellers ID
- Standard ID, e.g. the GS1 Global Trade Item Number (GTIN)

Which identifier to use depends on what is known at the time ordering or what is commonly used in the relevant business sector.



Each order line MUST have an item identifier and/or an item name

Example of an PEPPOL BIS Order item using both Sellers ID and Standard ID (GTIN):

```
<cac:SellersItemIdentification>
  <cbc:ID>SN-33</cbc:ID>
</cac:SellersItemIdentification>

<cac:StandardItemIdentification>
  <cbc:ID schemeID="0160">05704066204093</cbc:ID>
</cac:StandardItemIdentification>
```

XML

6.4. Product name and description

The Product name shall be sent in tag `cac:Item/cbc:Name` on line level. Description of a product can be sent in `cac:Item/cbc:Description`.

The Product name is often sent in the order from buyer to seller.

Example in an PEPPOL BIS Order message:

```
<cac:Item>
  <cbc:Description>1x12 pack sauce bags</cbc:Description>
  <cbc:Name>White sauce</cbc:Name>

  <!-- Code omitted for clarity -->
```

XML

6.5. Quantities and units

Various quantities and units can be stated in the PEPPOL BIS Order. These are both related to the ordering process and the logistics process.

The table below lists quantities and units in the format. To all quantities there must be a valid Unit according to the Code list.

Element name / (Tag name)	Description
Price Quantity (cbc:BaseQuantity)	Quantity related to Price.
Order Quantity (cbc:Quantity)	Quantity that is ordered, e.g. number of pieces or volume in litre .

Example of an order line with a quantity of 120 litre (cbc:Quantity) and price is given per litre.

```
<cbc:Quantity unitCode="LTR">120</cbc:Quantity>
<cac:Price>
  <cbc:PriceAmount currencyID="EUR">6</cbc:PriceAmount>
  <cbc:BaseQuantity unitCode="LTR">1</cbc:BaseQuantity>
</cac:Price>
```

XML

6.6. Prices

Prices may be exchanged in the order both for products with or without item identifiers and free text orders.

If prices are not sent in the order the normal process is to do price matching during the billing process comparing prices in the Invoice to prices in the Catalogue.

Price sent is related to the articles or services within this order.

- Prices should include allowances and/or charges but exclude VAT amounts

Example of price information in an Order message:

```
<cac:Price>
  <cbc:PriceAmount currencyID="EUR">30</cbc:PriceAmount>
  <cbc:BaseQuantity unitCode="NAR">2</cbc:BaseQuantity>
</cac:Price>
```

XML

6.7. Allowances and charges

6.7.1. General rules

Elements for allowance and charges are found on two levels:

1. The header level applies to the whole order and is included in the calculation of the order total amount.
2. The line level Price element. Data on this level may be provided to inform the buyer how the price is calculated. The price itself is always the net price, i.e. the Price Amount reduced/increased with allowancecharge/amount.

- Several allowances and charges may be supplied on header-level. For the Price-element only one occurrence of the Allowance and Charge element can be present. The element AllowanceCharge with sub element ChargeIndicator indicates whether the instance is a charge (true) or an allowance (false).
- The sum of all allowances and charges on the header level must be specified in AllowanceTotalAmount and ChargeTotalAmount respectively.
- Allowances and charges related to Price is for information only and not part of any other calculations.
- Allowances and charges related to Price may specify amount (AllowanceCharge/Amount) and base amount

6.7.2. Allowance and Charges on header level.

Example showing a charge related to packing costs:

```
<cac:AllowanceCharge>
  <cbc:ChargeIndicator>true</cbc:ChargeIndicator>
  <cbc:AllowanceChargeReason>Packing cost</cbc:AllowanceChargeReason>
  <cbc:Amount currencyID="EUR">100</cbc:Amount>
  <cac:TaxCategory>
    <cbc:ID>S</cbc:ID>
    <cbc:Percent>25</cbc:Percent>
    <cac:TaxScheme>
      <cbc:ID>VAT</cbc:ID>
    </cac:TaxScheme>
  </cac:TaxCategory>
</cac:AllowanceCharge>
```

XML

Example showing an allowance related to a discount on the order:

```
<cac:AllowanceCharge>
  <cbc:ChargeIndicator>false</cbc:ChargeIndicator>
  <cbc:AllowanceChargeReason>Agreed discount</cbc:AllowanceChargeReason>
  <cbc:Amount currencyID="EUR">100</cbc:Amount>
  <cac:TaxCategory>
    <cbc:ID>S</cbc:ID>
    <cbc:Percent>25</cbc:Percent>
    <cac:TaxScheme>
      <cbc:ID>VAT</cbc:ID>
    </cac:TaxScheme>
  </cac:TaxCategory>
</cac:AllowanceCharge>
```

XML

6.7.3. Allowance and Charges related to price

Example showing a discount on price of EUR 10:

```
<cac:Price>
  <cbc:PriceAmount currencyID="EUR">40</cbc:PriceAmount>
  <cac:AllowanceCharge>
    <cbc:ChargeIndicator>false</cbc:ChargeIndicator>
    <cbc:Amount currencyID="EUR">10</cbc:Amount>
    <cbc:BaseAmount currencyID="EUR">50</cbc:BaseAmount>
  </cac:AllowanceCharge>
</cac:Price>
```

XML

6.8. Calculation of totals (AnticipatedMonetaryTotals)

The following elements show the anticipated monetary totals for an order:

Element	Description	Formula

Element	Description	Formula
<cbc:LineExtensionAmount>	Sum of line amounts	$\sum (\text{cac:OrderLine}/\text{cac:LineItem}/\text{cbc:LineExtensionAmount})$
<cbc:AllowanceTotalAmount>	Allowances on document level	$\sum (\text{cac:AllowanceCharge}[\text{ChargeIndicator}=\text{'true'}]/\text{cbc:Amount})$
<cbc:ChargeTotalAmount>	Charges on document level	$\sum (\text{cac:AllowanceCharge}[\text{ChargeIndicator}=\text{'false'}]/\text{cbc:Amount})$
<cbc:TaxExclusiveAmount>	Order total amount without VAT	$\text{cac:LegalMonetaryTotal}/\text{cbc:LineExtensionAmount}$ – $\text{cac:LegalMonetaryTotal}/\text{cbc:AllowanceTotalAmount}$ + $\text{cac:LegalMonetaryTotal}/\text{cbc:ChargeTotalAmount}$
<cbc:TaxInclusiveAmount>	Order total amount included VAT	$\text{cac:LegalMonetaryTotal}/\text{cbc:TaxExclusiveAmount}$ + $\text{cac:TaxTotal}/\text{cbc:TaxAmount}$
<cbc:PrepaidAmount>	Any amounts that have been paid a-priory	<i>Not applicable</i>
<cbc:PayableRoundingAmount>	Rounding of Order total	<i>Not applicable</i>
<cbc:PayableAmount>	The amount that is expected to be paid	$\text{cac:LegalMonetaryTotal}/\text{cbc:TaxInclusiveAmount}$ – $\text{cac:LegalMonetaryTotal}/\text{cbc:PrepaidAmount}$ + $\text{cac:LegalMonetaryTotal}/\text{cbc:PayableRoundingAmount}$

- Amounts MUST be given to a precision of two decimals except for Price where maximum number of decimals are four.
- Expected total payable amount MUST NOT be negative.
- Expected total sum of line amounts MUST NOT be negative.



Note that the AnticipatedMonetaryTotals class is optional. If the class is included in the message, the only mandatory elements are the `cbc:LineExtensionAmount` and the `cbc:PayableAmount` elements. All other elements are optional. When optional elements are used, the content MUST be according to the formulas in the table above.*

6.8.1. Element for rounding amount, the PayableRoundingAmount

It is possible to round the expected payable amount. The rule for this is according to the standard rule regarding rounding, ie. greater than or equal to 0.5 is rounded up, all other values are rounded down.

The element `cac:AnticipatedMonetaryTotal/cbc:PayableRoundingAmount` is used for this purpose and is specified on the header level.

Example: Amount 999.81 rounded to 1000. PayableRounding Amount = 0.19.

6.8.2. Example of calculations:

Description	Element	Sample amounts
Sum of line amounts	<code>cbc:LineExtensionAmount</code>	700
Allowance on document level	<code>cbc:AllowanceTotalAmount</code>	100.00
Charges on document level	<code>cbc:ChargeTotalAmount</code>	200.00
Order total amount without VAT	<code>cbc:TaxExclusiveAmount</code>	800
VAT total amount	<code>cac:TaxTotal/cbc:TaxAmount</code>	85.63
Rounding of Order total	<code>cbc:PayableRoundingAmount</code>	0.37
Order total with VAT (value of purchase)	<code>cbc:TaxInclusiveAmount</code>	885.63
Paid amounts	<code>cbc:PrepaidAmount</code>	135.00
Amount expected to be paid	<code>cbc:PayableAmount</code>	751.00

The above example is presented in the order in the following way:

```

<cac:TaxTotal>
  <cbc:TaxAmount currencyID="EUR">85.63</cbc:TaxAmount>
</cac:TaxTotal>

<cac:AnticipatedMonetaryTotal>
  <cbc:LineExtensionAmount currencyID="EUR">700</cbc:LineExtensionAmount>
  <cbc:TaxExclusiveAmount currencyID="EUR">800</cbc:TaxExclusiveAmount>
  <cbc:TaxInclusiveAmount currencyID="EUR">885.63</cbc:TaxInclusiveAmount>
  <cbc:AllowanceTotalAmount currencyID="EUR">100</cbc:AllowanceTotalAmount>
  <cbc:ChargeTotalAmount currencyID="EUR">200</cbc:ChargeTotalAmount>
  <cbc:PrepaidAmount currencyID="EUR">135</cbc:PrepaidAmount>
  <cbc:PayableRoundingAmount currencyID="EUR">0.37</cbc:PayableRoundingAmount>
  <cbc:PayableAmount currencyID="EUR">751.00</cbc:PayableAmount>
</cac:AnticipatedMonetaryTotal>

```

XML

6.9. Tax total

It is possible to state the tax amount of the order, on the header level (tax total) and also on line level.

Example of tax total (header level)

```

<cac:TaxTotal>
  <cbc:TaxAmount currencyID="EUR">268.75</cbc:TaxAmount>
</cac:TaxTotal>

```

XML

6.10. Line VAT Category

VAT information on line level is provided in the class `cac:ClassifiedTaxCategory`.

If the class is used, each line shall have the item VAT category code, and for all VAT categories except "Not subject to VAT" (Category code O), the VAT rate shall be provided.

UBL example of line VAT category

```
<cac:ClassifiedTaxCategory>
  <cbc:ID>S</cbc:ID> 1
  <cbc:Percent>18</cbc:Percent> 2
  <cac:TaxScheme>
    <cbc:ID>VAT</cbc:ID> 3
  </cac:TaxScheme>
</cac:ClassifiedTaxCategory>
```

XML

¹ VAT category according to codelist [Code list UNCL5305](http://test-docs.peppol.eu/poacc/upgrade-3/codelist/UNCL5305/) (<http://test-docs.peppol.eu/poacc/upgrade-3/codelist/UNCL5305/>)

² VAT percentage, must be present unless VAT category code is O."Out of scope for VAT")

³ Value must be VAT

7. Peppol Identifiers

PEPPOL has defined a [PEPPOL Policy for identifiers](#).

(https://joinup.ec.europa.eu/svn/peppol/TransportInfrastructure/PEPPOL_Policy%20for%20use%20of%20identifiers-300.pdf) that specifies how to use identifiers in both its transport infrastructure and within the documents exchanged across that infrastructure. It also introduces principles for any identifiers used in the PEPPOL environment. The policies that apply to this BIS are the following:

7.1. Profiles and messages

All messages contains ProfileID and CustomizationID. ProfileID identifies what business process a given message is part of, and CustomizationID identifies the kind of message and the rules applied.

Profiles are connected to one business process, and may contain multiple document types. Valid document instances shall contain corresponding ProfileID and CustomizationID.



CustomizationID is a string without spaces. The list below contains spaces in CustomizationID to make them easier to read. **Make sure to remove any spaces before use.**

7.2. Customization and Profile identifiers

In the table below you will find the values to be used as the specification identifier and the business process type for this profile

Type	Element cbc:CustomizationID	Element cbc:ProfileID
Order (Trdm01)	urn:fdc:peppol.eu:poacc:trns01a:order:bis3-1.0	urn:fdc:peppol.eu:poacc:bis03a:order_only:bis3-1.0

7.3. Namespaces

- The target namespace for the [UBL 2.1](#) (<http://docs.oasis-open.org/ubl/UBL-2.1.html>) Order is:
`urn:oasis:names:specification:ubl:schema:xsd:Order-2`

Version 3.0.0

Last updated 2018-10-10 13:52:48 UTC