

PINT specialization guide

Table of Contents

Introduction.....	3
Architecture.....	4
Shared content.....	4
Aligned content.....	5
Distinct content.....	5
Global interoperability.....	6
Glossary and terminology.....	6
Derived specifications.....	7
Data model.....	8
Specializations.....	8
Allowed alignments.....	8
Cardinality of shared business terms.....	8
Distinct content.....	8
Possible additions.....	8
Limitations.....	8
Functionality.....	8
Tax inclusive pricing.....	9
Activating tax inclusive pricing.....	9
Changes in semantic meaning.....	10
Split payments.....	11
Example.....	12
Withholding tax.....	13
Payment terms.....	13
Line order and dispatch advice reference.....	13
Rules.....	13
Numbering of rules.....	13
Shared rules.....	14
Aligned rules.....	14
Distinct rules.....	14
Linking rules to syntax and business terms.....	14
Codes.....	16
Identifiers.....	16
Specification identifier (customization ID).....	16
PINT specification identifier scheme.....	16
Meaning of specification identifier.....	17
C4 receiving capability vs processing capability.....	17

Business process type (profile ID)	18
DDTS	18
Busdox	18
Busdox identifier scheme	18
Document Type Identifier	18
Further restrictions	19
Related processes	19
List of document identifications	20
Distinct business terms	21
Network exchange	21
Receiving capability	21
Setting up a PINT specialization for a new jurisdiction	22
Instructions for PINT	22
Phase one, establishing receiving capability	23
Phase two, establishing sending capabilities	23
C2 processing	24
Copying the PINT model to a jurisdiction	24
Localizing the specification	25
Applying alignments	25
Restrictions to cardinality	25
Example	26
Adding distinct content	27
Implementation	27
Migration	27
Potential starting cases include:	27
Potential end case (mandatory)	27
Mandating	27
Optional use	27
Maintenance	27
Developing and building locally	27
Syntax binding	27
Instructions for writing in adoc	28
Setting up Docker	28
Building locally	28
Sharing files between models	28
BIS document	28
Text formats	28
Sample image	29
Test snippet	30
Snippets	30
Insert a section from a XML file	31

Direct insert	32
Images	32
Alignment points	32

This is a guideline on how the PINT invoice specification can be specialized. The guideline is mainly intended for Peppol Authorities that want to create specializations that support requirements that are specific to their jurisdiction. The guideline addresses both the specializations that are allowed by the PINT methodology as well as giving instructions on how these specialization can be developed using the Peppol Development Toolkit (PDK tool).

This documents is not intended for supporting implementation of the resulting specification. For that purpose parties should use the relevant BIS specification. To avoid misunderstanding by the implementors of a specialization on how they must and may implement it, this guide should not be included with the published specialization.

Introduction

The entry of Singapore as the first non-EU PEPPOL Authority demonstrates that the current mandatory PEPPOL invoice as specified in the PEPPOL BIS Billing 3.0, which is based on the EN16931, poses challenges to countries that do not have VAT and/or countries that not in scope for EU tax regulations. The development of the EN16931 was mandated by Directive 2014/55/EU which states that the EN16931 must support EU Directive 2006/112 on VAT, which it does by applying mandatory rules and restricting possible tax schemes to VAT.

This poses the following challenges for PEPPOL.

- The PEPPOL mandatory principle is implemented for the invoice with the BIS Billing 3.0 specification which is a compliant implementation of the EN16931. As such it enforces EU tax rules which may not be relevant for non-EU countries that are joining PEPPOL.
- The current mandatory BIS Billing 3.0 for invoicing does not provide support for relevant regulation in the non-EU countries in PEPPOL.
- To maintain interoperability throughout its network, PEPPOL needs to develop an invoicing model that can be accepted as a mandatory specification under any legal environment.

To address these challenges, the OpenPEPPOL Post Award Community has, in coordination with OpenPEPPOL management, initiated a workgroup to produce a pre-study report. This pre-study report seeks to identify requirements and concerns related to invoicing in PEPPOL, in a wider international scope than within the European Union (EU) , based on input from the countries and regions participating in the workgroup. The report should produce recommendations on how to resolve these challenges. These recommendations will be submitted to OpenPEPPOL management who will then decide on further actions.

The workgroup should reach out and consult volunteer experts from different countries and regions with the purpose of gathering requirements. It should evaluate whether the development of an international invoicing model is feasible and make recommendations to the OpenPEPPOL

management on further activities.

The workgroup further defined the objectives of its work and identified general issues that should be taken into consideration.

- As legislation in different countries will change over time and introduce previously unidentified legal requirements, the invoicing model must allow for flexibility to quickly support such changes.
- An invoicing model should provide for completeness in supporting domestic requirements within and outside of the EU. It should provide for clarity in specifications to avoid ambiguity in content interpretations and enable high quality of invoice data on the network.
- International interoperability of invoices needs to be implemented with consideration to the identification of receiving capabilities within the PEPPOL network.
- An invoicing model should be practical and prioritise the ability to start implementing the solutions and then adjust based on feedback.

Architecture

The following diagram shows the three main components of the PINT data model.

This approach proposes a strictly defined small core that supports general business requirements and then a more generically defined layer that can be specialized for specific implementations, maintaining alignment. For example, instead of defining VAT it would define Tax that can be implemented as either VAT or GST. Any such specialization does not change the international invoice model itself but creates a specification that is compliant to the model itself. Specifications adding distinct content also do not change the international invoice model itself. This has similarities to the EN 16931 where the emphasis is that the core provides elements for all legal requirements and most business requirements allowing most implementations to be done by using restrictions. Further descriptions of each part of the model are provided below.

Shared content

The shared content of the model is the key for interoperability. It is intended to enable exchange of invoices across countries and other invoicing domains in a way that can be processed automatically by the receiver, while it does not necessarily fully support all requirements for the sender. The main characteristics of the shared content are the following:

- It is defined and used in the same way in all invoicing domains.
- It applies minimum rules to the content.
- It is enough for basic automations
 - Reading into ERP system.
 - Booking into accounts.
 - Order to invoice matching.

- Examples of content
 - Invoice meta data
 - Trading parties
 - Total amounts.
 - Items and prices.
 - References.

Aligned content

The aligned content is defined in a generalized way allowing it to be specialized in each invoicing domain. This allows the receivers to understand the received data in general terms but not necessarily specifically enough to automate its processing. The main characteristics of the aligned content are:

- It is defined in general terms but is expected to be given a more specialized definition in different invoicing domain.
- It can be understood in general terms by all domains.
- It contains no business rules, but rules can be added as part of the specialization.
- The generalized definition of the requirements is not aimed to support automation of processing although some automation may be achieved.
- Examples of content.
 - Tax information.
 - Payment instructions.

Distinct content

The model recognizes that in some invoicing domains some distinct content may be required, for different reasons. The international invoice model should not define these requirements since by doing so they would become either shared or aligned.

- The distinct content may not necessarily be understood by a receiver in a different invoicing domain.
- Examples of content.
 - Content that is distinct for different domains.
 - invoice domain specific legislation and practices.
 - sectoral legislation and practices.

<https://github.com/OpenPEPPOL/PINT/issues/5>

Global interoperability

Description of how PINT can be used

<https://github.com/OpenPEPPOL/PINT/issues/470>

<https://github.com/OpenPEPPOL/PINT/issues/267>

Glossary and terminology

Term	Definition
Invoicing domain	<p>Domains have specific characteristics in terms of legislation and requirements and may be defined by legal jurisdiction or industries. When trading partners are in the same domain, they share these characteristics but if they are in different domains, they may only share some of the characteristics.</p> <p>In international trade, the trading parties are in separate legal jurisdictions but may belong to the same industry.</p>
Invoice content	The content of an invoice, which serves the purpose of fulfilling the requirements that the invoice has been defined for.
Shared content	A set of business terms in the International semantic model, that are defined in detail so that they can be automatically processed without further specification and which definitions are shared unchanged by all users of the international semantic model.
Aligned content	A set of business terms in the International semantic model, that are defined in general terms and that encompass the specific meaning required by different invoicing domains, allowing for specialization to support the requirements of each invoicing domain.
Distinct content	Business terms that can be added to a specification but are not defined in the International invoice model.
Compliant	Some or all features of the international invoice model are used in the specification, and all rules are respected .
Conformant	All rules of the international invoice model are respected in the specification, and some additional features not defined in the invoice model are also used .
EN16931 (EN)	The European standard for electronic invoicing published by the European Standardization committee, CEN.
PEPPOL mandatory principle	Receivers with a registered receive capability for a business function for which a PEPPOL BIS is available shall have receive capabilities for the PEPPOL BIS registered in an SMP, as a minimum .
BIS	PEPPOL Business Interoperability Specification.

Term	Definition
SMP	Service Metadata Publisher. A service metadata publisher offers a service on the network where information about services of specific businesses can be found and retrieved. It is necessary for a client application to retrieve the metadata about the services for a target business before the client can use those services to send messages to the business.
SBDH	Standard Business Document Header can provide semantic information needed for the routing, processing and business domain context of documents, regardless of the data format of the document. Also called message envelope.
UBL Invoice	An XML messages syntax for an invoice published by Oasis. UBL is an abbreviation for Universal Business Language.
CII	An XML messages syntax for an invoice published by UNCEFACT. CII is an abbreviation for Cross Industry Invoice.
Indirect taxes	An indirect tax (such as sales tax, per unit tax, value added tax (VAT), or goods and services tax (GST), excise, tariff) is a tax collected by an intermediary (such as a retail store) from the person who bears the ultimate economic burden of the tax (such as the consumer). The intermediary later files a tax return and forwards the tax proceeds to government with the return.
VAT	Value Added Tax. The abbreviation is used as a general term for the value added tax system but when relevant and not otherwise stated in this document it refers to value added tax as applied in the European Union, based on the EU Directive 112/2006 and supported by the European eInvoicing Standard EN 16931.
GST	Goods and Service Tax. In this document it is used as a general term.
Sales tax	Sales tax. In this document it is used as a general term.
Dynamic receiving capability	A receiving capability that is registered in the SMP by using a wildcard and thus enabling reception of further specializations of the stated one.
Fixed receiving capability	A receiving capability in the SMP which only enables receptions of documents that have the exact customization id as is stated in SMP.
Processing capability	The receivers capability for processing a received message.

Derived specifications

Specifications that use the PINT general data model but do not comply with the shared business terms or rules. Examples are invoices for different functions such as pro-forma, self billing.

The PINT methodology or the PDK tools do not provide a defined method for building such specification in a way that they inherit from the general PINT invoice. Main challenge is that it depends on the function of the specification what is to be inherited and what not. There is no set rule on how to define that and defining it could go against the purpose.

However the import functionality of the in the PDK tool can be setup more granular then is used, which call for more complex setup but may enable such selected imports. This guide, however, makes not attempt to describe how this could be done.

Data model

Specializations

<https://github.com/OpenPEPPOL/PINT/issues/51>

<https://github.com/OpenPEPPOL/PINT/issues/411>

Allowed alignments

Cardinality of shared business terms

When shared business terms are optional (cardinality 0..1 or 0..n) then the receiver must handle the content of the business term if it is provided but he can not depend on the content to be provided. The reason why the sender provides the business term is irrelevant to the receiver.

It is therefore allowed to modify the cardinality of a shared optional business terms in a specialization so that it is made mandatory for the sender to provide it. This means that the following specializations are allowed:

- 0..1 may be changed to 1..1
- 0..n may be changed to 1..n

Distinct content

Possible additions

Limitations

Possible limitations

Functionality

The PINT data model supports various functionality which may be excluded in specializations depending on business practices and legislation in the relevant jurisdiction. This section describes the functionalities as they can be used.

A list of suggested features to explain is given in these issues:

<https://github.com/OpenPEPPOL/PINT/issues/466>

<https://github.com/OpenPEPPOL/PINT/issues/12>

Tax inclusive pricing

The PINT datamodel supports tax inclusive pricing as an optional functionality. The default functionality is tax exclusive using the same calculations as defined in the EN 16931 for document totals.

This enables specialization that support tax exclusive pricing only or both tax exclusive and tax inclusive. Specialization for tax inclusive only are not supported.

IBT-200 - Tax inclusive pricing, Aligned, 0..1

Indicates that item prices, allowances, charges or line total amounts are inclusive of tax.

This is a boolean element with allowed values "false" or "true".

(Since IBT-200 is boolean it has only three states, non-existing, existing as false or existing as true.)

Default value is false, meaning that if the business term does not exist in an invoice then, when relevant, its value is considered to be "false"

The schematron for the shared rules ibr-co-13 and ibr-co-15 have been modified by adding the precondition IBT-200 = false

The effect is that if IBT-200 does not exist or exists with the value "false" then the rules are run as before.

However if an invoice contains IBT-200 with value "true" then these rules are skipped (since IBT-200 is boolean it has only three state, non-existing, existing as false or existing as true.)

Specializations that do not support tax inclusive pricing (which includes those that are compliant to the EN 16931:2017) will set the cardinality for IBT-200 as 0..0. This excludes it and senders who comply with the specification can not include this business term in the invoices that they create.

EN 16931 validation artefact do not include the pre-condition that is added to the two rules in PINT. However since EN16931 compliand invoices will never contain IBT-200 they will pass both the EN 16931 validation artefacts and the PINT validation artefacts. When tested with EN 16931 the rules will a applied as normal, when tested with PINT the pre-condition will interpret the non-existance as false and run the same rules.

Activating tax inclusive pricing.

To activate the option of tax inclusive pricing the cardinality of IBT-200 is kept as 0..1.

The then need to add two new rules which are provided as rules ibr-co-27 and ibr-co-28 in the pooled rules.

These two rules are triggerd if IBT-200 exists with value "true". These rules calculate the values of IBT-109 Invoice total amount without TAX and IBT-112 Invoice total amount with TAX in revers.

When an invoice is tax exclusive then:

IBT-109 = sum of line amounts and allowance/charges on document level.
 IBT-112 = IBT-109 + IBT-110 Invoice total TAX amount

When an invoice is tax inclusive then:

IBT-112 = sum of line amounts and allowance/charges on document level.
 IBT-109 = IBT-112 - IBT-110 Invoice total TAX amount.

If a specialization applies rules for calculating tax then these rules need to be have preconditions for the possible states of the tax inclusive indicator and applying different calculation for each case.

For false the calculation is of the type:

$$\text{Tax} = \text{TaxExclusiveBaseAmount} \times \text{TaxRate}.$$

For true the calculation is on the line of

$$\text{TaxInclusiveBaseAmount} \times \text{TaxRate} / (1 - \text{TaxRate})$$

With tax inclusive pricing the LineAmount is TaxExclusive. When invoices are tax inclusive then Allowance and Charges must also be stated as tax inclusive. Consequently the taxable amount pr. rate will be inclusive of tax that rate.

Changes in semantic meaning

ID	Term name	Semantic definition	
		If false	If true
IBT-106	Sum of Invoice line net amount	Sum of all Invoice line net amounts in the Invoice.	The word net can not mean without tax.
IBT-107	Sum of allowances on document level	Sum of all allowances on document level in the Invoice.	No change needed but means that it includes tax.
IBT-108	Sum of charges on document level	Sum of all charges on document level in the Invoice.	No change needed but means that it includes tax.
IBT-109	Invoice total amount without TAX	The total amount of the Invoice without TAX.	Same
IBT-110	Invoice total TAX amount	The total TAX amount for the Invoice.	Same
IBT-112	Invoice total amount with TAX	The total amount of the Invoice with tax.	Same
IBT-113	Paid amount	The sum of amounts which have been paid in advance.	Same

IBT-114	Rounding amount	The amount to be added to the invoice total to round the amount to be paid.	Same
IBT-115	Amount due for payment	The outstanding amount that is requested to be paid.	Same
IBT-099	Document level charge amount	The amount of a charge, without TAX.	The amount of a charge, with TAX.
IBT-092	Document level allowance amount	The amount of an allowance, without TAX.	The amount of an allowance, with TAX.
IBT-131	Invoice line net amount	The total amount of the Invoice line (before tax).	The total amount of the Invoice line (with tax).
IBT-146	Item net price	The price of an item, exclusive of TAX, after subtracting item price discount.	The price of an item, inclusive of TAX, after subtracting item price discount.
IBT-141	Invoice line charge amount	The amount of a charge, without TAX.	he amount of a charge, with TAX.
IBT-136	Invoice line allowance amount	The amount of an allowance, without TAX.	The amount of an allowance, with TAX.

Split payments

The concept of split payments is generalized as any case when an invoice is settled in other ways than paying its full amount in a single payment. Use cases include the following:

- Seller allows the buyer to pay the amount of the invoice in multiple installments.
- Buyer is instructed to withhold the tax of the invoice and only pay the invoice amount without tax to the seller.

Split payments are handled with a combination of payment terms and payment means. In the payment terms the amount of the invoice can be split up and each amount given its separate due date and its separate payment means. In the payment means group each way of providing payment is defined and given an id.

ID	Name	Level	Crd	Description
IBG-33	INVOICE TERMS	Aligned	0..n	Information about the terms that apply to the settlement of the invoice amount.
IBT-187	Terms payment instructions ID	Aligned	0..1	An identifiers of the payment instructions that shall be used to settle the terms amount stated in (ibt-176).

ID	Name	Level	Crd	Description
IBT-020	Payment terms	Shared	0..1	A textual description of the payment terms that apply to the amount due for payment (Including description of possible penalties).
IBT-176	Terms amount	Aligned	0..1	The payment amount that these terms apply to. When relevant, the amount includes tax.
IBT-177	Terms installment due date	Aligned	0..1	The date before end of which the terms amount shall be settled.
IBG-16	PAYMENT INSTRUCTIONS	Aligned	0..n	A group of business terms providing information about the payment.
IBT-178	Payment Instructions ID	Aligned	0..1	An identifier for the payment instructions.
IBT-081	Payment means type code	Aligned	1..1	The means, expressed as code, for how a payment is expected to be or has been settled.
IBT-082	Payment means text	Shared	0..1	A textual description of the means/methods by which the payment is expected to be or has been settled.
IBT-083	Remittance information	Aligned	0..n	A textual value used for payment routing or to establish a link between the payment and the Invoice.
IBG-17	CREDIT TRANSFER	Aligned	0..1	A group of business terms to specify credit transfer payments.
IBG-18	PAYMENT CARD INFORMATION	Shared	0..1	A group of business terms providing information about card used for payment contemporaneous with invoice issuance.
IBG-19	DIRECT DEBIT	Aligned	0..1	A group of business terms to specify a direct debit.

Example

Payer may pay in separate installments with two different due dates. The first payment is by mutually defined means and the second is by bank transfer. The total amount of the invoice due for payment is €1000, first installment is €600 with due date 2013-10-01 and second is €400 with due date 2013-11-01.

[.xml] Example 1

```
<cac:PaymentTerms>
  <cbc:PaymentMeansID>A</cbc:PaymentMeansID>
  <cbc:Amount currencyID="EUR">600</cbc:Amount>
  <cbc:InstallmentDueDate>2013-10-01</cbc:InstallmentDueDate>
</cac:PaymentTerms>
<cac:PaymentTerms>
  <cbc:PaymentMeansID>B</cbc:PaymentMeansID>
  <cbc:Amount currencyID="EUR">400</cbc:Amount>
  <cbc:InstallmentDueDate>2013-11-01</cbc:InstallmentDueDate>
</cac:PaymentTerms>
...
<cac:PaymentMeans>
  <cbc:ID>A</cbc:ID>
  <cbc:PaymentMeansCode>ZZZ</cbc:PaymentMeansCode>
</cac:PaymentMeans>
<cac:PaymentMeans>
  <cbc:ID>B</cbc:ID>
  <cbc:PaymentMeansCode>30</cbc:PaymentMeansCode>
  <cac:PayeeFinancialAccount>
    <cbc:ID schemeID="IBAN">EU1234856789</cbc:ID>
  </cac:PayeeFinancialAccount>
</cac:PaymentMeans>
```

Withholding tax

When part of the invoice amount due is to be withheld due to tax the withholding is defined as a payment means with a suitable code.

Payment terms

<https://github.com/OpenPEPPOL/PINT/issues/61>

Line order and dispatch advice reference

<https://github.com/OpenPEPPOL/PINT/issues/145>

Rules

When using aligned rules the id of the aligned rule can only be used unchanged if the rule is not changed in any way. If there is any change made to the rule it needs to be post-fixed with the country code.

Numbering of rules

Shared rules

Shared rules have numbers that follow the format

- ibr-gg-*nnn*

where "ibr-" is a fixed prefix indicating that it is an internationally shared business rule, gg- is an optional group for the rule and *nnn* is a numerical sequence.

Aligned rules

All rules that are added to control the aligned content of PINT are called aligned rules. Aligned rules are those that are added to by the jurisdiction to enforce requirements that are specific to them. Recognizing that many of the jurisdictional requirements apply in multiple jurisdictions a pool of rules is provided from which a jurisdiction can draw rules for its own use.

The rules in the pool have the format

- aligned-ibrp-gg-*nnn*

where "aligned-ibrp" is a fixed prefix indicating that it is an aligned rule from the pool.

If a jurisdiction uses an aligned rule without making any changes to it it should keep the number of the rule unchanged. If a jurisdiction makes any change to the rule, such as changing values used in the rules or changing the message of the rules then the rule number shall be postfixed with the jurisdiction indicator, such as country code or region code. See example below.

- aligned-ibrp-gg-123-eu

If a jurisdiction writes its own aligned rule without drawing it from the pool the prefix of the rules is changed, dropping the p. See example below, and postfixing with the jurisdiction indicator. See example.

- aligned-ibr-gg-123-eu

Distinct rules

Distinct rules are only rules that act on distinct business terms. All rules that are restricting the specification of aligned business terms are called aligned rules irrelevant of whether they are drawn from the pool, modified from the pool or written from scratch by the jurisdiction. The numbering format of distinct rules is

- distinct-gg-*nnn*-*jj*

where gg distinct is a fixed prefix, gg is optional grouping, *nnn* is a numerical sequence and *jj* is the jurisdiction indicator.

Linking rules to syntax and business terms.

The key used for linking rules to business terms and syntax elements is the business term reference

ID (ibt-).

All semantic business terms have a reference ID and the bound syntax element will use the same reference unless it does not have a directly related semantic term, in which case it will have its own reference.

All rules have their own identifiers and SHALL in their message reference all semantic business terms and/or syntax elements that they act on.

The rules are linked to the business terms in the semantic file using the following syntax

Syntax for entering rule relations

```
id: ibt-001
name: Semantic Business term name
definition: Semantic business term description.
cardinality: Semantic cardinality
datatype: Semantic data type
section: Aligned
rule:
- id: The ID of a rule that acts on this business term.
  relatesTo:
  - The ID's of other business terms that relate to the rule.
  schematron:
  - The ID of the schematron rule
```

The following example is for a theoretical case where there is a business rule BR-001 that states that the business term "Total amount" that has ID ibt-001 must be the sum of amounts A and B, business terms ibt-002 and ibt-003 respectively. The implementation of rule BR-001 is supported by a specific technical syntax rule that has ID SR-001. In PINT Invoice such rules are not used even though the PDK tool supports that.

Example

```
id: ibt-001
name: Total amount
definition: The sum of amount A (ibt-002) and B (ibt-003)
cardinality: 1..1
datatype: Amount
section: Aligned
rule:
- id: BR-001
  relatesTo:
  - ibt-002
  - ibt-003
  schematron:
  - SR-001
```

Jurisdictions can add references to their jurisdictional rules (aligned and distinct content) by adding the rule references to the semantic business term in the following file

```
src\_common\alignments\data model\semantic_overwrites.yaml file.
```

All rule ID need to unique so there is not needt to reference the actual schematron file. Uniqueness of the rule IDs between jurisdiction is ensured by using the juristiction code in the rule id. There is not a fixed structure for this but one example could be br-001-jp for a japanese jurisdictional rule.

Codes

Consider <https://github.com/OpenPEPPOL/PINT/issues/421>

If a coded element has an assigned code list then that code list must be supported in full for shared elements but may be restricted for aligned elements.

All shared elements shall have a code list and that code list may not be extended or restricted.

If an aligned element has a code list then that list may be restricted but not extended or replaced. If the aligned element does not have an assigned code list then any code list may be assigned in a specialisation.

Identifiers

Specification identifier (customization ID)

PINT specification identifier scheme

Specification identifiers used in PINT are designed for the dynamic network functionality and are structured in the following way.

urn:peppol:pint:X@Y@Z ...

Where

- urn states that the id is a unique resource number
- peppol states that it the specification is designed for use in the Peppol network
- pint states that the design of the specification complies with the PINT methodology
- X is the general shared specification.
- Y identifies a specialization which complies with the xxx specification.
- Z is a further specialization of yyy wich complies with yyy and xxx, and so forth.

PINT billing document root identifier is

```
urn:peppol:pint:billing-1
```

Where urn:peppol:pint is states that this is a document specification from Peppol based on the PINT methodology. Following that billing-1 states that this is a billing document of version 1.0

As example for EU

```
urn:peppol:pint:billing-1@eu-1
```

The restricted version for each jurisdiction is identified with a code recognizing the jurisdiction, separated with the @ sign. There is not a set scheme for the format of these identifiers but recommended to keep them short.

Meaning of specification identifier

When stated in the CustomizationID element (ibt-024) inside and XML document it declares that the document complies with the full set of specifications expressed in the identifier.

As example, a XML document with identifier urn:peppol:pint:billing-1@en16931-2017@eu-3 is valid against the rules of the general PINT specification, the rules of the EN16931-2017 eInvoice standard and also all rules added by OpenPeppol in its Peppol BIS Billing 3.0 CIUS.

When stated as a receiving capability as part of the Document Identifier (eDec identifier) the customizationID declares the receivers (C4) receiving capability as well as his processing capability.

C4 receiving capability vs processing capability

As noted above the CustomizationID (as part of the DocumentID) states both C4 receiving and processing capabilities.

The PINT methodology allows using a wildcard in the SMP. When a DocumentTypeID is registered with a wildcard (*) in the SMP that states that the receiving capability is wider than the processing capability.

As example:

- A SMP receiving capability registration of urn:peppol:pint:billing-1@en16931-2017@eu-3 states that the receiver will receive a XML document with this customizationID and that he will also process it according to the full set of specification.
- A SMP receiving capability registration of urn:peppol:pint:billing-1@en16931-2017@eu-3* states that the receiver will receive a XML document with thsi customizationID and also XML documents that have customization ID that identify further specializations such as urn:peppol:pint:billing-1@en16931-2017@eu-3@ZZZ. However the receiver will only process this XML document according to the stated specifcation, i.e. he will not consider the specializations defined in ZZZ
- A SMP receiving capability registration of the general PINT specification, urn:peppol:pint:billing-1* states that the receiver will receive any specialization of the stated id, including XML documents with customizationID urn:peppol:pint:billing-1@en16931-2017@eu-3 but he will only process them according to the general PINT shared rules and will not consider what is definded by en16931-2017 or eu-3.

- A receiver with two receiving capability registrations in a SMP `urn:peppol:pint:billing-1*` and `urn:peppol:pint:billing-1@en16931-2017@eu-3*` is stating that he will receive any document that has a customizationID starting with `urn:peppol:pint:billing-1*` and will process it only according to the general PINT specification, except if the document has a customization ID starting with `urn:peppol:pint:billing-1@en16931-2017@eu-3` in which case he will process it according to that specialization. If he would receive an invoice from Japan with CustomizationID `urn:peppol:pint:billing-1@jp-1` he would process that according to the general pint specification only and ignore any specializations identified with `jp-1`.

A receiver (C4) should not enable the reception of a foreign invoice by registering its CustomizationID in the SMP unless he is also ready to process the invoice according to that country's specialization.

Business process type (profile ID)

Process identifiers identify the process that is supported by the BIS document.

<https://github.com/OpenPEPPOL/PINT/issues/452>

DDTS

`urn:peppol:bis:billing`

Identifies the billing process.

Busdox

When a PINT based invoice is exchanged through using the Busdox network exchange the Business process type can be the same as for DDTS. In the case of Japan the following is used for legacy reasons.

```
urn:fdc:peppol.eu:2017:poacc:billing:01:1.0
```

Busdox identifier scheme

If a PINT based document is exchanged using Busdox network functionality the following identifier is used, using Japan as an example. This identifies the same specification as `urn:peppol:pint:billing-1@jp` but follows the busdox identification scheme.

```
urn:fdc:peppol:jp:billing:3.0
```

Document Type Identifier

A document type identifier is used in the messaging network both in the SBDH message envelope and for registering receiving capabilities in the SMP.

The Document Type Identifier is constructed of several components, one of which is the CustomizationID.

The Document Type Identifier structure is as follows:

```
DocumentTypeIdentifierScheme::SyntaxSpecificID##CustomizationID::SyntaxVersion
```

There are two Document Type Identifier Scheme:

- busdox-docid-qns (BUSDOX)
- peppol-doctype-wildcard (DDTS)

They control the algorithm used to match the document to the receiving capability.

In general terms the busdox looks for an exact match, whereas DDTS uses a wildcard and looks for first match through a sequence. The wildcard is only used when registering the receiving capability, not in the document.

The wildcard matching algorithm requires the CustomizationID to follow the defined structure. On the other hand, since Busdox looks for an exact match it uses the CustomizationID as a fixed value and therefore its structure does not matter.

Consequently CustomizationID that support DDTS can be used with Busdox but not vice versa.

Further restrictions

A specification can be further restricted e.g. to support industrial requirements, by adding an identification for the restriction at the end of the identifier, e.g. as follows for

```
urn:peppol:pint:billing-1@eu@utilities
```

Which would identify an utility invoice that is a restriction of the European invoice (which is a restriction on the PINT invoice).

Related processes

Other processes can be defined based on a base process such as a PINT billing invoice.

Since a BIS identifies a process and because a receiver who can receive a document using one process may not necessarily be able or willing to receive the same document following a different process, then the identification of such related processes differ.

As an example, a self billing invoice process and invoice document may be defined based on the standard invoice BIS. Even though they are closely related there are two important differences. First is that the process is not the same. The self billing invoice is sent from the buyer to the seller. That means that a party who has a receiving capability for an invoice, when he is in the role of a buyer, may not be able to process an invoice in the role of a seller. When receiving an invoice in the role of a buyer he would read that invoice into his accounts as a purchase, whereas if receiving a

self billing invoice he would book it as sales revenue.

Additionally the self billing invoice may have slight difference in the rules that apply and possibly in the data model.

This means that neither the business process and the document specifications are the same and thus need to be identified differently.

Using the japanese invoice as an example. If a self billing process would be generalized it could be identified in a similar way as standar billing process but using a different process name, e.g. as follows:

```
urn:peppol:pint:selfbilling-1
```

The document specification identifier could have a generic component that could then be localized e.g. as follows

```
urn:peppol:pint:selfbilling-1
```

This specification could then be localized as follows, using Japan and AUNZ as examples.

```
urn:peppol:pint:selfbilling-1@jp  
urn:peppol:pint:selfbilling-1@aunz
```

List of document identifications

Document	Business process type	Specification identifier
Interoperable PINT BIS Billing specifications for different jurisdictions		
EU invoice	urn:peppol:bis:billing	urn:peppol:pint:billing-1@en16931-2017@eu-3
Singapore invoice	urn:peppol:bis:billing	urn:peppol:pint:billing-1@sg-1
AUNZ invoice	urn:peppol:bis:billing	urn:peppol:pint:billing-1@aunz-1
Japan invoice	urn:peppol:bis:billing	urn:peppol:pint:billing-1@jp-1
Malaysian invoice	urn:peppol:bis:billing	urn:peppol:pint:billing-1@my-1
Other PINT based invoice specifications		
Japan non-tax invoice	urn:peppol:bis:billing	urn:peppol:pint:nontaxinvoice-1@jp-1
Japan selfbilling invoice	urn:peppol:bis:selfbilling	urn:peppol:pint:selfbilling-1@jp-1

Document	Business process type	Specification identifier
AUNZ selfbilling invoice	urn:peppol:bis:selfbilling	urn:peppol:pint:selfbilling-1@aunz-1
Japan invoice for busdcox (for backwards compatibility)	urn:fdc:peppol.eu:2017:poacc:billing:01:1.0	urn:fdc:peppol:jp:billing:3.0

Distinct business terms

Numbering convention distinct business terms are BTXX-*nnn* Where:

- BT is a case neutral prefix identifying that it is a business term number
- XX is a code identifying the jurisdiction for which the distinct content is created. It is commonly the 2 digit alpha-2 ISO 3166 country code but may be different and longer if the jurisdiction is a region other domain. It should be an alpha code and case neutral.
- the hyphen is a part of the id.
- *nnn* is a 3 digit numerical id, usually moving up from 001 but not required to be sequential.

Business terms groups are prefixed with BG and uses a two digit numeric id as BGXX-*nn*

Network exchange

Explanation of how the PINT datamodel works with the Peppol Network.

<https://github.com/OpenPEPPOL/PINT/issues/485>

<https://github.com/OpenPEPPOL/PINT/issues/465>

Receiving capability

The sender, C1, declares what specification was used to construct the messages that he is sending by inserting the relevant customization identifier.

If the sender creates an invoice based on the European specialization of PINT, which is a compliant implementation of the European standard EN16931 he should enter the CustomizationID as

pint:billing@en16931@eu

By doing that it declares that the invoice is compliant to the general PINT specification, The EN 16931 specification and the Peppol specific CIUS. The invoice will validate against the rules of any or all of these specifications.

A receiver may register in an SMP that he is able to receive and process invoices that comply with the specifications as indicated in the document type identifier. The receiver may register multiple receiving capabilities. The receiving capability indicates two things.

- What CustomizationID C4 will receive.

- According to what level of the specification he will process the invoice.

C4 can use a wildcard to indicate that he will receive any specializations of an invoice but only process it down to a certain level.

Examples:

Case	Receiving capability	Meaning
A	pint:billing@EU	Will only receive an EU invoice and process it according to all the rules.
B	pint:billing*	Will receive any specialization of the general PINT invoice but only process according to the general rules.
C	pint:billing@en16931*	Will receive an EU standard invoices and also any invoices that are specializations (CIUS) of the EU standard invoice including the Peppol CIUS. All invoices are processed according to the general pint specification and the en16931 standard specification. Any further specializations are ignored.
D	pint:billing@en16931	Will only receive EN standard invoice but no CIUS.
E	pint:billing@en16931@EU*	Will receive the Peppol CIUS of the EN 16931 only but no other CIUS nor full EN. Will also receive further specialization of the Peppol CIUS. All invoices will be processed according to the general PINT, EN 16931 and Peppol CIUS rules but any further specializations will be ignored.
D	pint:billing@en16931@EU*	Will receive the Peppol CIUS of the EN 16931 only but no other CIUS nor full EN. Will also receive further specialization of the Peppol CIUS. All invoices will be processed according to the general PINT, EN 16931 and Peppol CIUS rules but any further specializations will be ignored.

As example if C4 has registers receiving ca

Setting up a PINT specialization for a new jurisdiction

<https://github.com/OpenPEPPOL/PINT/issues/460>

Instructions for PINT

Following are instructions the necessary steps to activate PINT in a new jurisdiction

Phase one, establishing receiving capability.

1. The necessary new doc type id need to be created by eDec (OO)
 - a. urn:peppol:pint:billing-1@nn-1
 - i. where nn is the code identifying the jurisdiction specific specialization.
2. SMP operator needs to make sure he can register wildcard document types.
3. C4 may add PINT receiving capability with or without wildcard in the SMP. IF wildcard then he needs to have the appropriate receiving capability. IF not wildcard then he can use what he has, note difference in customizationID.
 - a. some C4 may accept wildcard, then using DocID:
 - i. peppol-doctype-wildcard:urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##urn:peppol:pint:billing-1@nn-1::2.1*
 - b. some C4 may not accept wildcard, then using DocID:
 - i. busdox-docid-qns:urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##urn:peppol:pint:billing-1@aunz-1::2.1
 - c. some C4 may have: peppol-doctype-wildcard:urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##urn:peppol:pint:billing-1@nn-1::2.1*

A Peppol jurisdiction (PA) may decide to allow only wildcard DocID as receiving capability but allowing only busdox is not possible.

Phase two, establishing sending capabilities.

1. C1 modifies XML to use PINT customization ID (urn:peppol:pint:billing-1@nn-1)
2. C2 adds functionality to resolve wildcard lookup.
 - a. When receiving from C1 a valid XML without a SBDH envelope.
 - i. Validate XML, success.
 - ii. Perform SMP lookup.
 - A. Pulls receiving capabilities
 - B. Looks match according endpoint selection parameters.
 - C. If no match, abort, else continue.
 - iii. Create SBDH, based on SMP lookup parameters.
 - iv. Sends with AS4.
 - b. When receiving from C1 a valid XML with a SBDH envelope.
 - i. Validate XML, success.
 - ii. Perform SMP lookup.
 - A. Pulls receiving capabilities.
 - B. Same lookup as before.

- C. If no match, abort, else continue.
- iii. Modify SBDH, based on SMP lookup parameters.
- iv. Sends with AS4.

Design principle is that pr. jurisdiction old XML = new XML (payload except customization ID), they should therefore be routed identically.

1. C1 can be ignorant of whether C4 accepts wildcard version or not.
2. SBDH is not dependent on previous lookup.
3. C4 need to be able to receive a particular custID irrelevant of what SBDH scheme is stated.
4. The overall concept is already challenging for understanding so we need to simplify and be logical.
5. The SBDH is the responsibility of C2, even though he may receive a preliminary one from C1.

C2 processing

- From C1, valid XML w/o SBDH
 1. Validate XML, success.
 2. Perform SMP lookup.
 - a. Pulls receiving capabilities
 - b. Looks match according endpoint selection parameters.
 - c. If no match, abort, else continue.
 3. Create SBDH, based on SMP lookup parameters.
 4. Sends with AS4
- From C1, valid XML with SBDH
 1. Validate XML, success.
 2. Perform SMP lookup.
 - a. Pulls receiving capabilities
 - b. Same lookup as before.
 - c. If no match, abort, else continue.
 3. Modify SBDH, based on SMP lookup parameters.
 4. Sends with AS4

Copying the PINT model to a jurisdiction

1. Create an new repository for the jurisdiction as a copy of the PINT repository.
2. Change name of directory for the BIS src/pint to postfix name as pint-xx where the xx is and indicator for the jurisdiction. As example. pint-eu for the EU pint.
3. In the pint-xx folder replace the .import.yaml file with the Generic.imports.yaml that is in the

src/_common folder. After the replace the name of the generic file shall be .imports.yaml

4. Delete folder LegacyMaterial
5. Delete folder ProjectDocuments
6. Delete folder RuleDev
7. Replace library.yaml with generic jurisdiction library

Localizing the specification

The new specification can be edited so that titles and text are relevant for the jurisdiction for which the specification is being developed.

The following files can be edited.

- Main page for the jurisdiction: project.yaml
- BIS main page: src\pint-XX\process.yaml
- BIS document title: src\pint\docs\bis\settings.adoc
- BIS document description: src\pint-XX\docs\bis\main.adoc
- Transaction name and description src\pint\trn-XX\transaction.yaml

Graphics can be added to some of the text by uploading to a folder and insert into the text using ascii doc commands.

Applying alignments

Alignments are changes that are within the scope of the PINT (restrictions) and may only be made to aligned part of the data model. No change may be made to the shared part. The following types of alignments may be made:

- Restricting the cardinality of the aligned part of the data model.
- Localizing the name and definition of a business term.
- Localizing selected code lists
- Adding localized sections into the BIS document.
- Add rules that do not conflict with shared rules.

Restrictions to cardinality

Restriction to cardinality are made by editing the following file by removing the hashtag (#) in front of the line that is to be changed and then applying the change. It is good practice to show the original in a comment at the end of the line.

```
src\_common\alignments\data model\semantic_overwrites.yaml
```

Cardinality controls how often a business term may appear in a data model. It is written as x..y

where x is the minimum and y is the maximum. The x and the y can take the values of 0, 1 and n where n stands for unlimited. It is possible to use other intergers but they should be avoided.

As example, an element with cardinality 0..1 may at minimum be excluded and at maximum be included once, cardinality 1..n may at minimum be included once and the maximum is unlimited. A common terminology is that business terms with a minimum of 0 are called optional and those with a minimum of 1 are called mandatory.

Following cardinality changes may be made.

- 0..y may be changed to
 - 1..y, which changes the minimum number of occurrence to one, making it mandatory for the sender to include the business term in the message.
 - 0..0, which changes the maximum number of occurrences to zero and thus removes the business term from the datamodel.
- x..n (0..n and 1..n) may be changed to
 - x..1, changing the maximum number of occurrence from unlimited to one.

Following cardinality changes may not be made in alignments. * A minimum (x value) may not be lowered. As example 1..1 may not be changed to 0..1 * A maximum (y value) may not be increased. As example 1..1 may not be changed to 1..n

Example

Following example shows where the sellers tax identifier is made mandatory and its names and description aligned to show the tax name GST.

Original:

```
# - id: ibt-031
# name: Seller TAX identifier
# definition: The Seller's TAX identifier (also known as Seller TAX identification
number).
# cardinality: 0..1
# codelist:
# example:
```

Aligned:

- id: ibt-031 # name: Seller TAX identifier # definition: The Seller's TAX identifier (also known as Seller TAX identification number). cardinality: 1..1 # codelist: # example:

Note that the hashtag must be removed from the id line as well as the space between the hashtag and the dash. It is also important that the following lines align with the id. The hashtag should not be removed from lines that are not being changed.

These cardinality changes will be automatically adopted into both the semantic and the syntax models. In the syntax model that may be by changing a higher level class instead of the directly

mapped element.

Adding distinct content

Implementation

Migration

Migration to PINT depend on the starting case and the end point

Potential starting cases include:

- i. There is no existing use of Peppol BIS billing specification in the jurisdiction. Example are new Peppol PA who start by using PINT.
- ii. There are existing users with receiving capability for a Peppol BIS using a non-PINT customization ID with fixed receiving capabilities (busdox). Examples are Singapore, Europe and Australia
- iii. There are existing users with receiving capability for a Peppol BIS using a PINT customizationID for fixed receiving capabilities (busdox). Example is Japan.

Potential end case (mandatory)

- α. All receivers only have dynamic receiving capability for pint*
- β. Receivers have either fixed or dynamic receiving capability for a specialization.
- γ. Add more

Mandating

Optional use

Maintenance

Developing and building locally

Overview

Syntax binding

<https://github.com/OpenPEPPOL/PINT/issues/344>

<https://github.com/OpenPEPPOL/PINT/issues/100>

Instructions for writing in adoc

<https://docs.asciidoctor.org/asciidoc/latest/>

Setting up Docker

Building locally

Main commands

```
.\docker-pdk.ps1
.\docker-pdk-serve
docker stop pdk-serve
.\docker-pdk.ps1 clean - cleans the local build to ensure full update. Useful to get
rid of deleted files.
docker ps - Shows running servers
tree
docker run -d -p 80:80 docker/getting-started
wsl --shutdown
```

1. Start from command prompt in root and run Powershell
2. Navigate to the directory where the jurisdiction is located
3. run `.\docker-pdk-serve` to initiate the local server.
4. run `.\docker-pdk.ps1` to build. The build finishes with table showing files build and should give a sound signal.
5. view local build on path `localhost:8000/` in a browser

Sharing files between models

Using `_common`, `_library` and `imports.yaml`

BIS document

BIS documents are written by using Ascii doc which is a form of a markdown language. Instructions can be found here: <https://docs.asciidoctor.org/asciidoc/latest/>

Selected functions that are used in the BIS documents are described below.

Text formats

Text is formatted by prefixing it with a different characters as follows

```
= Heading on level 1
== Heading on level 2
*bold*
_italic_
~subscript~
^superscript^
#highlight#
```

By putting one or more character space at the start of a line the text in that line is displayed without formatting or by activating a function.

Sample text for describing the use of the PINT in the context of the jurisdiction. Description may include:

- This
- That

In the order of

1. **first in bold**
2. *second in italics*

Sample image

Images can be inserted into the text by using the following function:

```
.function for inserting images
image::{image.jpg}[Imagename, width="400", align="center"]
```

When images are stored in a image folder (dir) the folder can be referred to in two ways. Either by using the parameter `{:imagesdir:}` to point to the relative location of them image and then insert the image name only or by leaving the imagedir empty and provide the image name with its relative path. As example, where file being inserted into is in root folder and images are in root/images

- Not using imagedir

```
in main.adoc
:imagesdir:
```

```
in page.adoc
image::{images/image.jpg}[Imagename, width="400", align="center"]
```

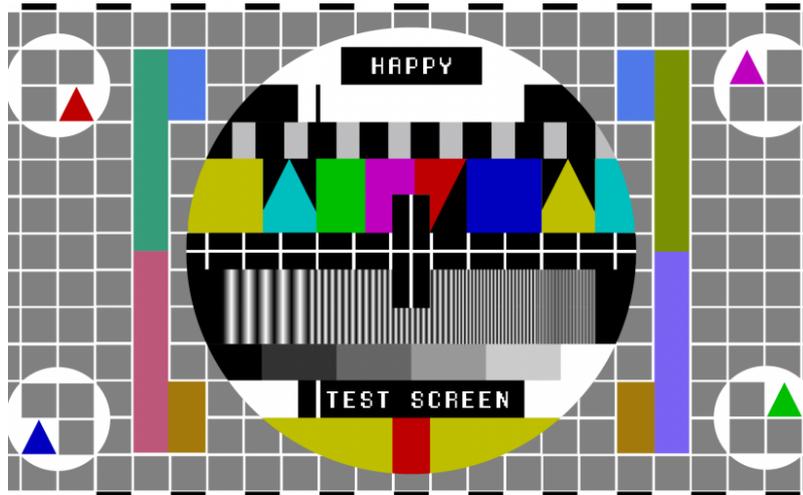
- Using imagedir

```
in main.adoc
:imagesdir: images
```

```
in page.adoc
image::{image.jpg}[Imagename, width="400", align="center"]
```

Which displays as follows:

When not using imagesdir



When using imagesdir = images and only image name

[Test image] | *Happy-Test-Screen.png*

The benefit of using the imagesdir parameter in main.adoc is that then images can be inserted by using their name only and the relative path is not needed for each image. The use of full relative path for each image has the benefit that then the image displays both in VirtualStudio preview and in the PDK build but when using imagesdir it only displays in the build.

The folder in which the image is stored must contain an empty text file named .adocassets

Test snippet

Following is an example of how a code snippet can be inserted into the aligned content as part of explanations. The example shows the XML for invoice id and issue date.

A heading for the code snippet

```
<cbc:ID>Snippet1</cbc:ID>
<cbc:IssueDate>2017-11-13</cbc:IssueDate>
```

Snippets

XML snippets can be inserted into the text for explanations and to give examples. There are two

methods that can be used to insert snippets.

Insert a section from a XML file

In this method an XML file is created separately and the part of the XML code that is to be displayed in the document is tagged in the XML file and then included into the document. The benefit of this approach is that the snippet can be taken from a full XML file that can be validated and used as a full example.

As an example. In the folder `src\pint\alignments\bis\snippets` the xml file has been created `Snippet-Full.xml`. In the top of the file a section has been tagged as follows using the tag "profile".

```
<?xml version="1.0" encoding="UTF-8"?>
<Invoice xmlns:cac=
"urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"
  xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
  xmlns="urn:oasis:names:specification:ubl:schema:xsd:Invoice-2">
  <!-- tag::profile[] -->

  <cbc:CustomizationID>urn:cen.eu:en16931:2017#compliant#urn:fdc:peppol.eu:2017:poacc:bi
lling:3.0</cbc:CustomizationID>
  <cbc:ProfileID>urn:fdc:peppol.eu:2017:poacc:billing:01:1.0</cbc:ProfileID>
  <!-- end::profile[] -->
  <cbc:ID>Snippet1</cbc:ID>
  <cbc:IssueDate>2017-11-13</cbc:IssueDate>
```

This snippet is then displayed here by using the following ascii doc using the tag `profile` to identify the xml to be shown.

```
.UBL example of BIS identifiers
[source, xml, indent=0]
----
include::Snippet-Full.xml[tags=profile]
----
```

This is then displayed as follows in the document

UBL example of BIS identifiers

```
<cbc:CustomizationID>urn:cen.eu:en16931:2017#compliant#urn:fdc:peppol.eu:2017:poacc:bi
lling:3.0</cbc:CustomizationID>
<cbc:ProfileID>urn:fdc:peppol.eu:2017:poacc:billing:01:1.0</cbc:ProfileID>
```

When using this method it is very important to use the correct path to the xml file. The path needs to give relative to the main ascii doc file. As example `../../folder/subfolder` if the relative path is two levels up and then down into folder/subfolder like `include::../../folder/subfolder/Snippet-Full.xml[tags=profile]`

Direct insert

It is also possible to insert the snippet directly into the document file. The benefit of this approach is simplicity.

```
.UBL example of BIS identifiers  
[source, xml]  
<cbc:CustomizationID>urn:cen.eu:en16931:2017#compliant#urn:fdc:peppol.eu:2017:poacc:bi  
lling:3.0</cbc:CustomizationID>  
<cbc:ProfileID>urn:fdc:peppol.eu:2017:poacc:billing:01:1.0</cbc:ProfileID>
```

Which displays as follows

UBL example of BIS identifiers

```
<cbc:CustomizationID>urn:cen.eu:en16931:2017#compliant#urn:fdc:peppol.eu:2017:poacc:bi  
lling:3.0</cbc:CustomizationID>  
<cbc:ProfileID>urn:fdc:peppol.eu:2017:poacc:billing:01:1.0</cbc:ProfileID>
```

Images

[Test image] | *peppol.jpg*

Alignment points

The main body of the BIS document describes the shared functionality of the invoice.

The main BIS document has alignment points that connect with document sections that can be filled out with explanations that are relevant for the relevant jurisdiction. These documents are found in folder src\pint\alignments\bis\doc-section.

The aligned content can be entered directly into these document are a new file created and included.