



The future is open  
**Peppol**

Peppol model for Tax Data Document

# Table of Contents

Link to main site of documentation .....	2
Introduction .....	3
Scope .....	3
Audience .....	3
Benefits .....	4
1. Business processes .....	5
1.1. Parties and roles .....	5
1.2. Tax Data Document process .....	6
Tax Data Document functionality .....	6
2. Rules .....	8
2.1. Custom content .....	8
2.2. Source document .....	8
3. Technical details .....	9
3.1. BIS Identifiers .....	9
3.2. Datatypes .....	9
3.3. XML schemas and namespaces .....	13
3.4. Glossary .....	14



# Link to main site of documentation

[Main documentation site](#)

# Introduction

The purpose of this document is to describe a Tax Data Document (TDD) which may be used to report issued and received invoices to a relevant authority for the purpose of VAT reporting using continuous transaction control (CTC) methods.

## Statement of copyright

This Peppol Business Interoperability Specification (Peppol BIS) document is a Country Specification based on the PINT. The restrictions on PINT implemented in this Peppol BIS are identified in the conformance statement provided in appendix A.

The copyright of PINT is owned by OpenPeppol and its members. OpenPeppol AISBL holds the copyright of this Peppol BIS.

This Peppol BIS document may not be modified, re-distribute, sold or repackaged in any other way without the prior consent of OpenPeppol AISBL.

## Scope

This document is concerned with clarifying the structure and functionality of the TDD transaction and how it can be applied in different use cases.

## Audience

The audience for this document is organisations wishing to be Peppol enabled for CTC tax reporting, and/or their ICT-suppliers. These organisations may be:

- Service providers
- Contracting Authorities (CA)
- Economic Operators (EO)
- Software Developers
- Government Authorities

More specifically, roles addressed are the following:

- ICT Architects
- ICT Developers
- Business Experts

For further information on Peppol/OpenPeppol see <https://peppol.org>

# Benefits

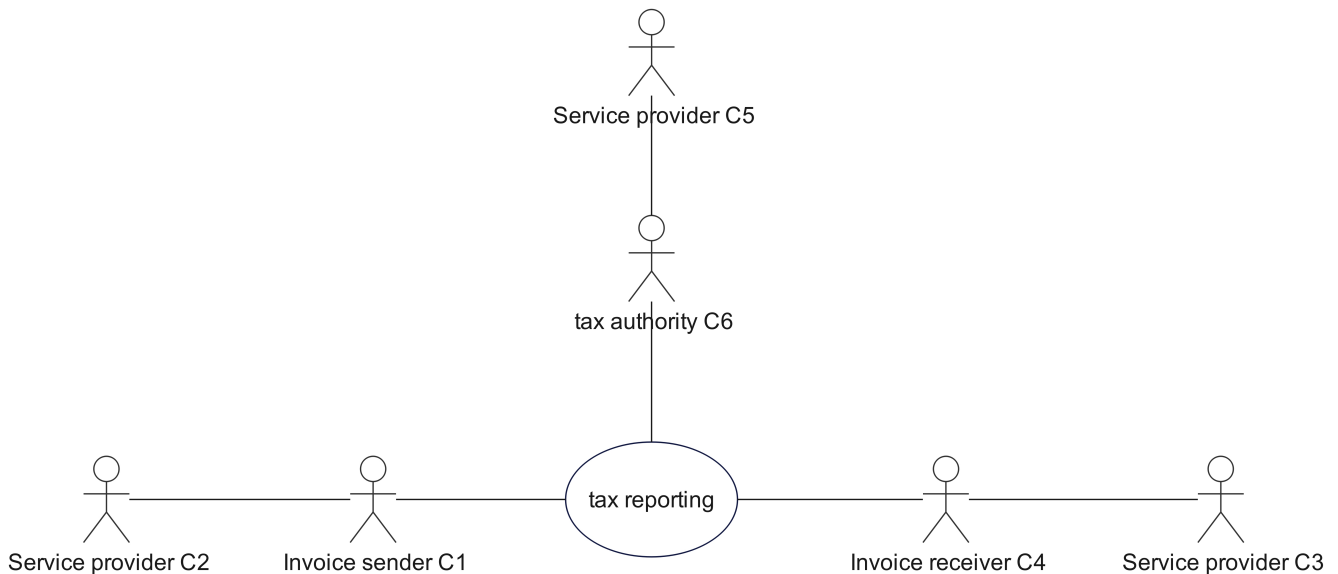
The TDD transaction may be used by either a seller or a buyer, or a service provider on their behalf, to report the sending or reception of an invoice to a relevant government authority, such as a tax authority, for the purpose of tax declaration.

- The TDD can be used by either the issuer or the receiver of an invoice.
- It provides information about the TDD message itself.
- It support management of the TDD message such as submit, replace, withdraw, that may be independent of the invoice exchange between the seller and the buyer.
- It provides for different levels of detail about the invoice ranging from simple extract of basic information from the invoice to full inclusion of the invoice.
- It allows for full inclusion of an invoice either as attachment or in expanded form.

# Chapter 1. Business processes

## 1.1. Parties and roles

The diagram below shows the roles involved in the invoice and credit note transactions. The customer and invoice receiver is the same entity, as is the supplier and the invoice sender.



### 1.1.1. Parties

#### Invoice sender C1

The party who, in the role of a supplier, issues an invoice to an invoice receiver. That invoice may contain tax which the invoice sender, in the role of a taxable person, is responsible for reporting to the relevant tax authority.

#### Service provider C2

A party who provides electronic messaging services that include preparing and transmitting invoices and TDD transactions on behalf of the invoice sender, C1.

#### Service provider C3

A party who provides electronic messaging services that include preparing and transmitting TDD transactions on behalf of C4 and receiving and forwarding invoice the invoice receiver, C4.

#### Invoice receiver C4

The legal person or organisation who, in the role of a buyer, is in demand of a product or service and in the role of a taxable person, is responsible for reporting to the relevant authority. Examples of invoice receiver roles: buyer, consignee, debtor, contracting authority.

#### Service provider C5

A party who provides electronic messaging services that include receiving and managing TDD transactions on behalf of a tax authority, C6

## Tax authority C6

A public authority that is responsible for collecting taxes and enforcing relevant legislation.

### 1.1.2. Roles

#### Taxable person

An organization or an individual who is responsible for reporting and settling taxes in accordance with applicable legislation.

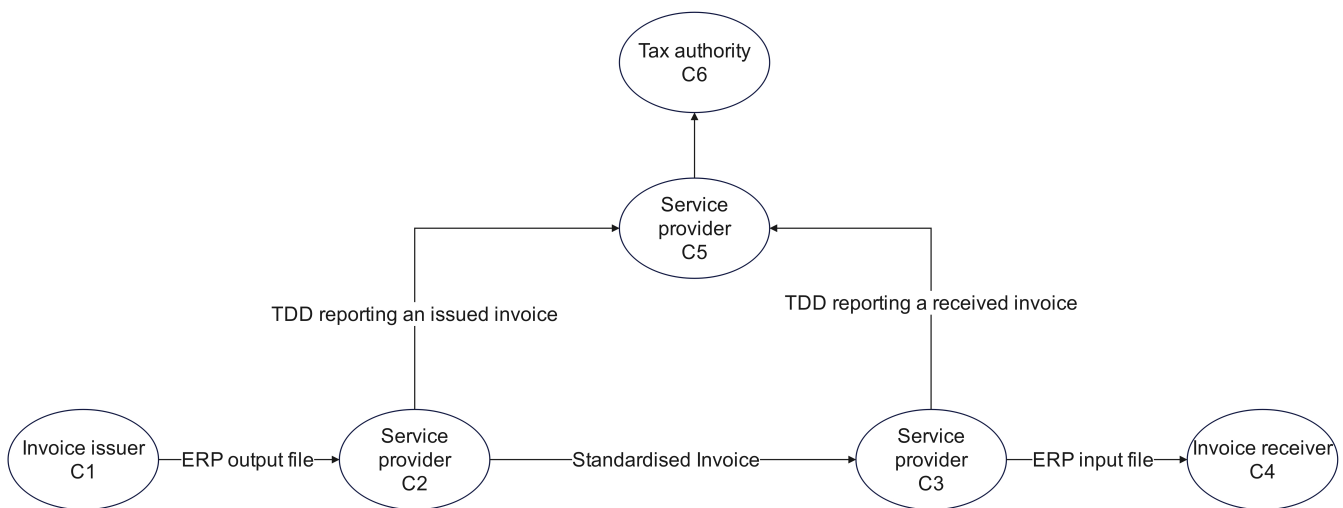
#### Service provider

A party who assists a another party in carrying out his responsibilities.

## 1.2. Tax Data Document process

The TDD process is dependent on the legislation of jurisdiction where it is being used and the objectives that the process is intended to achieve in that jurisdiction. This affect the content, the sequence and the timing of the TDD in the overall process.

The generic process can be described with the following diagram.



Depending on the implementation of TDD in the relevant jurisdiction process variation may include the following:

- The invoice receiver may not be required to report the reception of the invoice.
- The timing of forwarding of the invoice from C2 to C3 may be before or after the sending of the TDD. In a clearance model the the TDD would need to be sent and a response received from the tax authorities before the invoice can be forwarded but in a reporting model the sequence may be more varied.
- The exchange of the TDD may be independent of the invoice in the sense that correction and resending of the TDD may or may not require an equal exchange of the invoice.

## Tax Data Document functionality

The TDD document is contains three main sections:

- TDD document meta data
- Data extracts from the reported invoice.



- Inclusion of the full invoice.

## **TDD meta data**

The TDD is a report that is based on its own specification. It is exchanged with the tax authorities as opposed to between a buyer and a seller and it may be sent at a different time.

The first part of the TDD document contains metadata that identifies the specification, the relevant parties and dates and time. It also provides codes that define the role of the parties in the exchange and the function of each instance.

- A given party may send TDD's either in the role of an invoice issuer or invoice receiver.
- There may be more than one instance of a TDD for the same invoice that have different functionality such as submit, update and withdraw.

## **Data extracts**

One TDD instance may be used to report more than one invoice depending on the implementation in the relevant jurisdiction.

For each reported invoice key data values may be extracted into the TDD, depending on the relevant requirements.

## **Full invoice**

Where required the full reported invoice may be included in the TDD either as a digitally encoded attachment (base 64) or as extended XML.

Unresolved directive in doc-section/business-processes.adoc - include::../aligned-section/local-processes.adoc[]

# Chapter 2. Rules

The information given in a TDD must comply to a set of rules on the content of the business terms as well as the relationship between them.

## 2.1. Custom content

Description of how the custom content class shall be used.

## 2.2. Source document

Description of how source document (full invoice) shall be insterted when that is required.

# Chapter 3. Technical details

Following section provide technical details.

## 3.1. BIS Identifiers

Peppol has a defined policy 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:

### 3.1.1. Profiles and messages

All messages contains Business process type (IBT-23) and Specification identifier (IBT-24). Business process type (IBT-23) identifies what business process a given message is part of, and Specification identifier (IBT-24) 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 Business process type (IBT-23) and Specification identifier (IBT-24).



Specification identifier (IBT-24) is a string without spaces. The list below contains spaces in Specification identifier (IBT-24) to make them easier to read. **Make sure to remove any spaces before use.**

In the table below you will find the values to be used as the specification identifier (IBT-24) and the business process type (IBT-23) for this profile

Unresolved directive in doc-section/technical/bis-id.adoc - include::.../aligned-section/aligned-bis-id.adoc[]

*UBL example of profile identifier*

```
<cbc:CustomizationID>urn:peppol:pint:billing-1@specialization</cbc:CustomizationID>  
<cbc:ProfileID>urn:peppol:bis:billing</cbc:ProfileID>
```

### 3.1.2. Document type identifier scheme

Prior to 15 May 2025, exact match receiving capabilities will continue to use the busdox-docid-qns Document Type Identifier scheme. After 15 May 2025, exact match receiving capabilities will migrate to using the peppol-doctype-wildcard Document Type Identifier scheme, in accordance with the Peppol Policy for use of identifiers v4.3.0 migration plan.

## 3.2. Datatypes

Semantic data types are used to bridge the gap between the semantic concepts expressed by the information elements defined in the semantic model 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.

### 3.2.1. Primitive types

Semantic data type content may be of the following primitive types. These primitive types were taken from ISO15000, 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 ISO8601.
Decimal	A subset of the real numbers, which can be represented by decimal numerals.
String	A finite sequence of characters.

### 3.2.2. 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 {ISO15000}.

When used in an instance of an invoice, 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.

#### Amount

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

Component	Use	Primitive Type	Example
Content	Mandatory	Decimal	10000

#### Unit Price Amount

A unit 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

## 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

## 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

## 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 of the applicable syntax.

Component	Use	Primitive Type	Example
Content	Mandatory	String	Abc123

## Indicator

Indicators are used to give boolean values to state whether something is (true) or is not (false).



Indicators shall be used in lower case.



Default value is "false" and applies if the relevant business term is not used.

Component	Use	Primitive Type	Allowed values
Content	Mandatory	String	false
			true

## 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	Optional	String	GLN
Scheme version identifier	Optional	String	1.0

## Date

Dates shall be in accordance to the “Calendar date complete representation” as specified by {ISO8601}, 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

## Time

Time shall be according to UBL allowed format.



Time may include timezone information.

Table 2. EN 16931\_ Date. Type

Component	Use	Primitive Type	Allowed forms
Content	Mandatory	Date	13:20:00 (1:30 PM)
			13:20:30.55 (30.55 sec)
			13:20:00Z (UTC)
			13:20:00-05:00 (UTC-5)
			00:00:00 (midnight)
			24:00:00 (midnight)

Time formats without time zone information (i.e. other than hh:mm:ssz and hh:mm:ss-h) shall be

interpreted as being in the time zone of the sellers address and according to daylight saving status on the issue date of the invoice.

## Document Reference

Document Reference Types are identifiers that were assigned to a document or document line by the Buyer, the Seller or by a third party.

Table 3. Document Reference. Type

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

## 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

## Binary object

Binary object can be used to describe files which are transmitted together with the Invoice. The attachment functionality is not intended for attaching a copy of the invoice in an image format (such as PDF). Attaching an invoice copy is not in compliance with this specification.

Attachments shall be transmitted together with the Invoice. 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 invoice or credit note.

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

A receiver of an invoice or credit note, shall accept and process attachments that are according to the Media type code list.

## 3.3. XML schemas and namespaces

The primary XML schemas used is

- OpenPeppol TDD, with the target namespace `urn:peppol:schema:pint:peppol:tdd:1.0`

As this XML Schema is defined by OpenPeppol itself, it is handled slightly different as e.g. the

schema for UBL Invoice. The TDD XML Schema reuses as much as possible from OASIS UBL 2.1 common components but having its own primary identity.

Reused XML namespaces in the TDD XML schemas are:

Prefix	URL
<b>cac</b>	urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2
<b>cbc</b>	urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2
<b>cec</b>	urn:oasis:names:specification:ubl:schema:xsd:CommonExtensionComponents-2
<b>udt</b>	urn:oasis:names:specification:ubl:schema:xsd:UnqualifiedDataTypes-2

For technical reasons, there are no deep integrations with the required XML schemas but instead it is expected an [XML Catalog]([https://en.wikipedia.org/wiki/XML\\_catalog](https://en.wikipedia.org/wiki/XML_catalog)) is used to resolve the required XML schemas. The referenced XML schemas can be downloaded from the [OASIS UBL 2.1](<https://docs.oasis-open.org/ubl/os-UBL-2.1/>) publication.

## 3.4. Glossary

**electronic invoice** - invoice that has been issued, transmitted and received in a structured electronic format which allows for its automatic and electronic processing

**semantic data model** - structured set of logically interrelated information elements

**information element** - semantic concept that can be defined independent of any particular representation in a syntax

**structured information element** - information element that can be processed automatically

**syntax** - machine-readable language or dialect used to represent the information elements contained in an electronic document (e.g. an electronic invoice)

**business term** - label assigned to a given information element which is used as a primary reference

**identifier** - character string used to establish the identity of, and distinguish uniquely, one instance of an object within an identification scheme from all other objects within the same scheme Note 1 to entry: An identifier may be a word, number, letter, symbol, or any combination of those, depending on the identification scheme used.

**identification scheme** - collection of identifiers applicable for a given type of object governed under a common set of rules

**compliant** - some or all features of the invoice model are used and all rules of the invoice model are respected Note 1 to entry: Based on TOGAF definition of a compliant specification



**conformant** - all rules of the invoice model are respected and some additional features not defined in the invoice model are also used

**Optional** Whether the option is used or not is the choice of the users independently from other data in the message.

**Conditional** Whether the option is used or not and in what way is dependent on other data in the message.

**Mandatory** The option must be used in all messages.

**shall** - the definition is an absolute requirement of the specification.

**shall not** - the definition is an absolute prohibition of the specification.

**should** - there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

**should not** - there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

**may** - is truly optional.