



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	3
1. Business processes	5
1.1. Parties and roles	5
1.2. Tax Data Document process	6
1.3. Tax Data Document functionality	7
2. Technical details	8
2.1. BIS Identifiers	8
2.2. Datatypes	8
2.3. XML schemas and namespaces	12
2.4. Glossary	12

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, together with related metadata, to a relevant authority for the purpose of VAT reporting using Continuous Transaction Control (CTC) methods.

Statement of copyright

OpenPeppol AISBL holds the copyright of this Peppol specification. This Peppol BIS document may not be modified, re-distributed, 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 consists of organisations wishing to be Peppol-enabled for CTC tax reporting, and/or their ICT suppliers. These organisations may include:

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

More specifically, the roles addressed include 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 by a service provider acting on their behalf, to report the sending or receipt of an invoice to a relevant government authority, such as a tax authority, for the purpose of tax declaration.

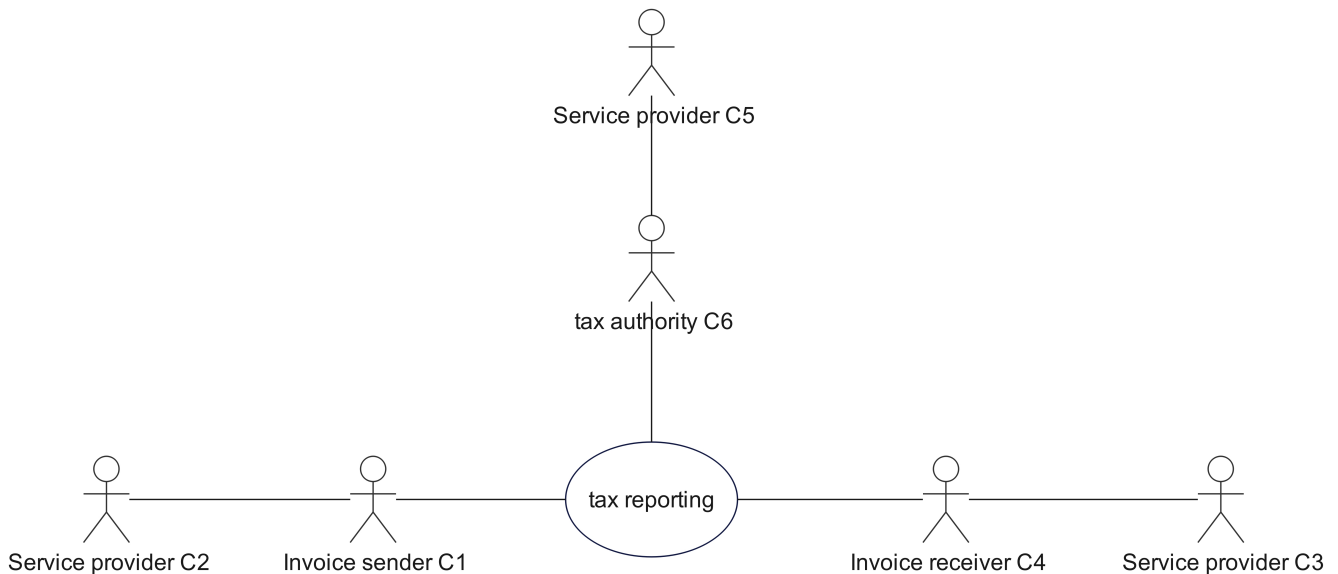
- The TDD can be used by either the issuer or the receiver of an invoice.
- It provides information about the TDD message itself.

- It supports management of the TDD message, such as **Submit**, **Resubmit**, and **Disregard**, which may be independent of the invoice exchange between the seller and the buyer.
- It provides different levels of detail about the invoice, ranging from a simple extract of basic invoice information to full inclusion of the invoice.
- It allows for full inclusion of an invoice, either as an attachment or in expanded form.

Chapter 1. Business processes

1.1. Parties and roles

The diagram below illustrates the roles involved in invoice and credit note transactions. The customer and the invoice receiver are the same entity, as are 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. The 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 that provides electronic messaging services, including preparing and transmitting invoices and TDD transactions on behalf of the invoice sender, C1.

Service provider C3

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

Invoice receiver C4

The legal person or organisation who, in the role of a buyer, demands 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 include buyer, consignee, debtor, and contracting authority.

Service provider C5

A party that provides electronic messaging services, including 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 another party in carrying out its responsibilities.

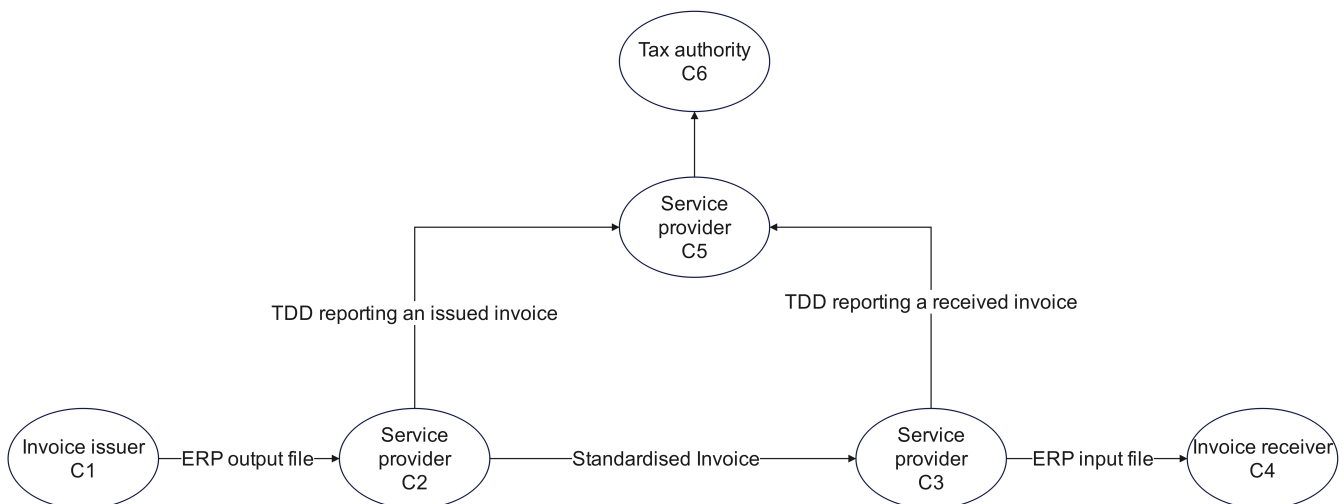
Tax authority

A public body or government organization responsible for administering tax legislation, receiving tax data documents, and assessing, collecting, and enforcing taxes.

1.2. Tax Data Document process

The TDD process depends on the legislation of the jurisdiction in which it is used and on the objectives the process is intended to achieve in that jurisdiction. This affects the content, sequence, and timing of the TDD within the overall process.

The generic process can be described using the following diagram.



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

- The invoice receiver may not be required to report the receipt of the invoice.
- The timing of forwarding the invoice from C2 to C3 may occur either before or after the sending of the TDD. In a clearance model, the TDD must be sent and a response received from the tax authorities before the invoice can be forwarded. In a reporting model, the sequence may be more flexible.
- 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 a corresponding exchange of the invoice.

1.3. Tax Data Document functionality

The information provided in a TDD must comply with a defined set of rules governing both the content of business terms and the relationships between them.

The TDD document contains two main sections:

- TDD document metadata
- Data extracts from the reported invoice

1.3.1. TDD metadata

The TDD is a report based on its own specification. It is exchanged with tax authorities, rather than 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 the applicable dates and times. It also provides codes that define the role of the parties involved in the exchange and the function of each TDD instance.

- A given party may send TDDs either in the role of an invoice issuer or an invoice receiver.
- There may be more than one TDD instance for the same invoice, with different functions such as **Submit**, **Resubmit**, and **Disregard**.

1.3.2. Data extracts

A single 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 applicable requirements.

Chapter 2. Technical details

Following section provide technical details.

2.1. 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:

2.1.1. 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 (TDT-001) and the business process type (TDT-002) for this profile

Type	Element <code>cbc:CustomizationID</code>	Element <code>cbc:ProfileID</code>
Tax Data Document	urn:peppol:taxdata:sk-1	urn:peppol:taxreporting

UBL example of profile identifier

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

2.1.2. Document type identifier scheme

The `busdoox-docid-qns` Document Type Identifier scheme shall be used in accordance with the current version of the Peppol Policy for the use of identifiers.

2.2. 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.

2.2.1. 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
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.

2.2.2. 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	10000.25

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 shall include time zone 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.

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

2.3. XML schemas and namespaces

The primary XML schema used is:

- OpenPeppol TDD, with the target namespace `urn:peppol:schema:sk-taxdata:1.0`

As this XML Schema is defined by OpenPeppol itself, it is handled slightly differently than, for example, the schema for the UBL Invoice. The TDD XML Schema reuses as many common components from OASIS UBL 2.1 as possible, while maintaining its own primary identity.

The XML namespaces used in the TDD XML Schema are:

Prefix	URL
pxs	<code>urn:peppol:schema:sk-taxdata:1.0</code>
cac	<code>urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2</code>
cbc	<code>urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2</code>
cec	<code>urn:oasis:names:specification:ubl:schema:xsd:CommonExtensionComponents-2</code>
udt	<code>urn:oasis:names:specification:ubl:schema:xsd:UnqualifiedDataTypes-2</code>

For technical reasons, there are no deep integrations with the required XML schemas. Instead, it is expected that an [XML Catalog](#) is used to resolve the required XML schemas. The referenced XML schemas can be downloaded from the [OASIS UBL 2.1](#) publication.

2.4. Glossary

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

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

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

message.

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

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

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

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.

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

Mandatory - The option must be used in all messages.

May - is truly optional.

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

Semantic data model - structured set of logically interrelated information elements

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.

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)