SLOVENSKI STANDARD
kSIST-TS FprCEN/TS 16931-3-2:2026

01-april-2026

Elektronsko izdajanje računov - 3-2. del: Povezava sintakse za račun in dobropis EN 16931-1 na ISO/IEC 19845 (UBL)

Electronic invoicing – Part 3-2 Syntax binding of EN 16931-1 to ISO/IEC 19845 (UBL) invoice and credit note

Ta slovenski standard je istoveten z: **FprCEN/TS 16931-3-2**

ICS:

03.100.20	Trgovina. Komercialna dejavnost. Trženje	Trade. Commercial function. Marketing
35.240.63	Uporabniške rešitve IT v trgovini	IT applications in trade

kSIST-TS FprCEN/TS 16931-3-2:2026 **en,fr,de**

TECHNICAL SPECIFICATION

SPÉCIFICATION TECHNIQUE

TECHNISCHE SPEZIFIKATION

FINAL DRAFT
FprCEN/TS 16931-3-2

February 2026

ICS

Will supersede CEN/TS 16931-3-2:2020

English Version

Electronic invoicing - Part 3-2 Syntax binding of EN 16931-1 to ISO/IEC 19845 (UBL) invoice and credit note

This draft Technical Specification is submitted to CEN members for Vote. It has been drawn up by the Technical Committee CEN/TC 434.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and United Kingdom.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Warning : This document is not a Technical Specification. It is distributed for review and comments. It is subject to change without notice and shall not be referred to as a Technical Specification.

EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

Ref. No. FprCEN/TS 16931-3-2:2026 E

Contents

Page

European foreword

This document (FprCEN/TS 16931-3-2:2026) has been prepared by Technical Committee CEN/TC 434 “Electronic invoicing”, the secretariat of which is held by NEN.

This document is currently submitted to the Vote on TS.

This document has been prepared under a mandate given to CEN by the European Commission and the European Free Trade Association, and supports essential requirements of EU Directive 2014/55/EU [1] and Council Directive 2006/112/EC [2].

This document supersedes CEN/TS 16931-3-2:2017.

This document is part of a set of documents, consisting of:

— EN 16931-1:2026 Electronic invoicing - Part 1: Semantic data model of the core elements of an electronic invoice

— CEN/TS 16931-2:2017, Electronic invoicing - Part 2: List of syntaxes that comply with EN 16931-1

— CEN/TS 16931-3-1:2026, Electronic invoicing - Part 3 - 1: Methodology for syntax bindings of the core elements of an electronic invoice

— CEN/TS 16931-3-2:2026, Electronic invoicing - Part 3 - 2: Syntax binding for ISO/IEC 19845 (UBL) invoice and credit note

— CEN/TS 16931-3-3:2026, Electronic invoicing - Part 3 - 3: Syntax binding for UN/CEFACT XML Cross Industry Invoice

— CEN/TS 16931-3-4:2026, Electronic invoicing - Part 3 - 4: Syntax binding for UN/EDIFACT INVOIC

— CEN/TR 16931-4:2017, Electronic invoicing - Part 4: Guidelines on interoperability of electronic invoices at the transmission level

— CEN/TS 16931-5:2026, Electronic invoicing - Part 5: Guidelines on the use of sector or country extensions in conjunction with EN 16931-1, including a methodology to be applied in the real environment

— CEN/TR 16931-6:2017, Electronic invoicing - Part 6: Result of the test of the European standard with respect to its practical application for an end user - Testing methodology

— CEN/TR 16931-7, Electronic invoicing - Part 7: Methodology for the development and use of EN 16931-1 compliant structured Core Invoice Usage Specifications

— CEN/TR 16931-8, Electronic invoicing - Part 8: Semantic data model of the elements of an e-receipt or a simplified electronic invoice

— CEN/TR 16931-9, Electronic invoicing - Part 9: VAT reporting and gap analysis with current e-invoicing standardization deliverables

— CEN/TR 16931-10, Electronic invoicing – Part 10: Additional requirements to extend to B2B

Any feedback and questions on this document should be directed to the users’ national standards body. A complete listing of these bodies can be found on the CEN website.

Introduction

The European Commission estimates that “The mass adoption of e-invoicing within the EU would lead to significant economic benefits and it is estimated that moving from paper to e-invoices will generate savings of around EUR 240 billion over a six-year period” [3]. Based on this recognition “The Commission wants to see e-invoicing become the predominant method of invoicing by 2020 in Europe.”

As a means to achieve this goal, Directive 2014/55/EU [1] on electronic invoicing in public procurement aims at facilitating the use of electronic invoices by economic operators when supplying goods, works and services to the public administration (B2G), as well as the support for trading between economic operators themselves (B2B). In particular, it sets out the legal framework for the establishment and adoption of a European standard (EN) for the semantic data model of the core elements of an electronic invoice (EN 16931-1).

The semantic data model of the core elements of an electronic invoice – the core invoice model – as described in EN 16931-1 is based on the proposition that a limited, but sufficient set of information elements can be defined that supports generally applicable invoice-related functionalities.

This CEN Technical Specification CEN/TS 16931-3-2 defines the binding of the core elements of the invoice to UBL. Other subparts of this CEN Technical Specification define the binding method (CEN/TS 16931-3-1) and map the core invoice model to other syntaxes such as UN/CEFACT XML (CEN/TS 16931-3-3) and UN/EDIFACT (CEN/TS 16931-3-4).

By ensuring interoperability of electronic invoices, the European standard and its ancillary European standardization deliverables will serve to remove market barriers and obstacles to trade deriving from the existence of different national rules and standards – and thus contribute to the goals set by the European Commission.

1 Scope

This document specifies the mapping between the semantic model of an electronic invoice, included in EN 16931-1 and the UBL syntax. For each element in the semantic model (including sub-elements or supplementary components such as Identification scheme identifiers) it is defined which element in the syntax is to be used to contain its information contents. Any mismatches between semantics, format, cardinality or structure are indicated.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

EN 16931-1, *Electronic invoicing - Part 1: Semantic data model of the core elements of an electronic invoice*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
Electronic Invoice
invoice that has been issued, transmitted and received in a structured electronic format which allows for its automatic and electronic processing

[SOURCE: Directive 2014/55/EU [1]]

3.2
Information Element
smallest unit of data that is used to represent an item of information within an Electronic Invoice

Note 1 to entry:
The EN identifies these elements using Business Terms (BTs). In EN 16931-1 section 6.3 is a table of information elements contained in the Core Invoice Model.

3.3
Structured Information Element
information element that can be processed automatically

3.4
Business Term
label assigned to a given information element which is used as a primary reference

3.5
Business Terms Group
group of related Business Terms

Note 1 to entry:
BTs can be aggregated within Business Terms Groups (BGs). For example, the BG Seller contains all the information elements needed to describe the entity that is selling the good or service. BG Seller also contains its own BGs such as address and contact i.e. BG Seller acts as a parent Group to child Groups for addresses and contact details that are related to the Seller.

3.6
Semantic Data Model
structured set of logically interrelated information

3.7
Core Invoice Model
semantic data model of the Core Elements of an Electronic Invoice

Note 1 to entry: The model contains the Core Elements of an Electronic Invoice – see EN 16931-1 Clause 4 for a more detailed explanation. The Core Invoice Model is composed of mandatory information elements that every invoice shall contain along with conditional elements that can be used when required.

3.8
Core Elements of an Electronic Invoice
set of information elements that most Electronic Invoices contain in order to enable interoperability, including the necessary information to ensure legal compliance

3.9
Extended Information Element
information element within the Scope for Extensions but outside the Core Invoice Model

Note 1 to entry: Extended Information Elements are sometimes informally referred to as extensions in other documents.

3.10
Core Invoice Usage Specification (CIUS)
specification that provides detailed guidance, explanations, and examples, as well as rules (business rules) related to the actual implementation and use of structured information elements present in the Core Invoice Model in a specific trading situation

3.11
Core Invoice Instance Document
instance of an Electronic Invoice that is conformant with the Core Invoice Model

3.12
Extension Specification
specification describing the use of Extended Information Elements to the Core Invoice Model that may reuse Extension Components

Note 1 to entry: An Extension Specification is intended to be published in the eInvoice Registry. It is typically written by a Representative/Representatives of a Sectoral Organisation for its members to describe an Invoice that includes the Core Semantic Model elements, Extension Components, and other elements needed for business.

Note 2 to entry: The resulting invoice model contains information elements that do not form a strict subset of the Core Invoice Model. An Extension Specification can also provide additional explanations and examples.

3.13
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.

3.14
Identification Scheme
collection of Identifiers applicable for a given type of object governed under a common set of rules

3.15
Compliant
meets all the legal requirements and follows all the legal rules of any Directive associated with the standard, particularly the VAT Directive

3.16
Core Conformant
respects all the normative rules of the Core Invoice Model

Note 1 to entry: A Core Conformant instance is not expected to throw any error when using CEN/TC434/WG3 validation artefacts for the Core Invoice Model.

3.17
Syntax
machine-readable format used to represent the information elements contained in an Electronic Invoice instance

Note 1 to entry: CEN/TS 16931-2 contains the list of syntaxes that comply with EN 16931-1 and that are mandatory for public bodies in the European Union.

4 Syntax binding to UBL

4.1 Introduction

The Universal Business Language (UBL) is developed by the OASIS open standards consortium. OASIS is a non-profit, international consortium that drives the development, convergence and adoption of open standards for the global information society.

UBL is designed to provide a universally understood and recognized syntax for legally binding business documents and to operate within a standard business framework such as ISO 15000 (ebXML)[4] to provide a complete, standards-based infrastructure that can extend the benefits of existing EDI systems to businesses of all sizes. UBL is freely available to everyone without legal encumbrance or licensing fees.

UBL is widely used around the world for procurement (e.g. ordering and electronic invoicing), sourcing (e.g. tendering and catalogues), replenishment (e.g. managed inventory) and transportation and logistics (e.g. waybills, forwarding instructions, and intermodal freight management). UBL provides the standards for the PEPPOL (Pan European eProcurement Online) network and public procurement initiatives in Austria, Belgium, Czech Republic, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Netherlands, Norway, Spain, Sweden, and UK (NHS).

UBL schemas are modular, reusable, and extensible in XML-aware ways. As the first standard implementation of ebXML Core Components Technical Specification 2.01, the UBL Library is based on a conceptual model of information components known as Business Information Entities (BIEs). These components are assembled into specific document models such as invoice and order. These document models are then transformed in accordance with UBL Naming and Design Rules into W3C XSD schema syntax as defined from the OASIS UBL Technical Committee [5]. This approach facilitates the creation of UBL-based document types beyond those specified in this release.

4.2 UBL XML versions

This Syntax binding is based on UBL Version 2.5. Although it is not restricted to a specific UBL version, only UBL 2.5 and any subsequent UBL 2.x version have all syntax elements which are needed for the current revision EN 16931-1:2026.

The old EN 16931-3-2:2017 was based on UBL 2.1 (released as an ISO and IEC International Standard, and given the designation 'ISO/IEC 19845:2015'). Later versions of UBL are backwards compatible with version 2.1.

The following Tables list the new UBL 2.5 syntax elements being used in EN 16931-3-2:2026 in the order of the semantic model. Table 1 shows the XPaths for the Invoice, Table 2 for the Credit Note. For guidance on backwards compatible mapping, see Tables 14 and 15:

Table 1 — New UBL invoice XML

ID	XPath (old and new)	Reason
BT-10	**UBL2.1**: /Invoice/c**b**c:BuyerReference **Since UBL 2.5**: /Invoice/c**a**c:BuyerReference/c**b**c:BuyerReference	BT-10 changed to have unbounded cardinality (appear more than once) and may have a Code in BT-10-1
BT-10-1	**Since UBL 2.5** /Invoice/cac:BuyerReference/cbc:BuyerReferenceCode	Qualifying the Buyer Reference with a Code from UNTDID 1153 [6]
BT-197	**Since UBL 2.5**: /Invoice/cac:DeliveryNoteDocumentReference/cbc:ID	New BT
BT-182	**Since UBL 2.5**: /Invoice/cac:PaymentTerms/cac:PenaltyInterestRate/cbc:InterestRatePercent	New BT
BT-21	**UBL 2.1**: /Invoice/cbc:Note **Since UBL 2.5**: /Invoice/cac:Annotation/cbc:SubjectCode	There was no Invoice Note Subject Code in UBL 2.1
BT-22	**UBL 2.1**: /Invoice/cbc:Note **Since UBL 2.5**: /Invoice/cac:Annotation/cbc:AnnotationContent	Invoice Note could be added in UBL 2.1 only once. Unbounded were its translations.
BT-213	**Since UBL 2.5**: /Invoice/cac:AllowanceCharge/cac:TaxCategory/cbc:SupplyTypeCode	New BT with cbc:ChargeIndicator [.='false']
BT-214	**Since UBL 2.5**: /Invoice/cac:AllowanceCharge/cac:TaxCategory/cbc:SupplyTypeCode	New BT with cbc:ChargeIndicator [.='true']

ID	XPath (old and new)	Reason
BT-179	**Since UBL 2.5**: /Invoice/cac:CollectionInvoiceLine/cbc:TaxInclusiveLineExtensionAmount	New BT
BT-180	**Since UBL 2.5**: /Invoice/cac:CollectionInvoiceLine/cac:Item/cbc:Description	New BT
BT-210	**Since UBL 2.5**: /Invoice/cac:TaxTotal/cac:TaxSubtotal/cac:TaxCategory/cbc:SupplyTypeCode	New BT
BT-198	**Since UBL 2.5**: /Invoice/cac:InvoiceLine/cac:Delivery/cac:DeliveryNoteDocumentReference/cbc:ID	New BT
BT-199	**Since UBL 2.5**: /Invoice/cac:InvoiceLine/cac:Delivery/cac:DeliveryNoteLineReference/cbc:LineID	New BT
BT-196	**Since UBL 2.4**: /Invoice/cac:InvoiceLine/cac:Item/cac:ClassifiedTaxCategory/cbc:ItemSupplyTypeCode	New BT

Table 2 — New UBL credit note XML

ID	XPath (old and new)	Reason
BT-9	**UBL 2.1**: /CreditNote/cac:PaymentMeans/cbc:PaymentDueDate **Since UBL 2.2**: /CreditNote/cbc:DueDate	There was no CreditNote Note cbc:DueDate in UBL 2.1
BT-10	**UBL 2.1**: /CreditNote/c**b**c:BuyerReference **Since UBL 2.5**: /CreditNote/c**a**c:BuyerReference/c**b**c:BuyerReference	BT-10 changed to have unbounded cardinality (appear more than once) and may have a Code from UNTDID 1153 [6]
BT-10-1	**Since UBL 2.5** /CreditNote/cac:BuyerReference/cbc:BuyerReferenceCode	Qualifying the Buyer Reference with a Code from UNTDID 1153 [6]
BT-197	**Since UBL 2.5**: /CreditNote/cac:DeliveryNoteDocumentReference/cbc:ID	New BT

ID	XPath (old and new)	Reason
BT-182	**Since UBL 2.5**: /CreditNote/cac:PaymentTerms/cac:PenaltyInterestRate/cbc:InterestRatePercent	New BT
BT-21	**UBL 2.1**: /CreditNote/cbc:Note **Since UBL 2.5**: /CreditNote/cac:Annotation/cbc:SubjectCode	There was no Credit Note Subject Code in UBL 2.1
BT-22	**UBL 2.1**: /CreditNote/cbc:Note **Since UBL 2.5**: /CreditNote/cac:Annotation/cbc:AnnotationContent	Credit Note could be added in UBL 2.1 only once. Unbounded were its translations.
BT-213	**Since UBL 2.5**: /CreditNote/cac:AllowanceCharge/cac:TaxCategory/cbc:SupplyTypeCode	New BT with cbc:ChargeIndicator [.='false']
BT-214	**Since UBL 2.5**: /CreditNote/cac:AllowanceCharge/cac:TaxCategory/cbc:SupplyTypeCode	New BT with cbc:ChargeIndicator [.='true']
BT-179	**Since UBL 2.5**: /CreditNote/cac:CollectionCreditNoteLine/cbc:TaxInclusiveLineExtensionAmount	New BT
BT-180	**Since UBL 2.5**: /CreditNote/cac:CollectionCreditNoteLine/cac:Item/cbc:Description	New BT
BT-210	**Since UBL 2.5**: /CreditNote/cac:TaxTotal/cac:TaxSubtotal/cac:TaxCategory/cbc:SupplyTypeCode	New BT
BT-198	**Since UBL 2.5**: /CreditNote/cac:CreditNoteLine/cac:Delivery/cac:DeliveryNoteDocumentReference/cbc:ID	New BT
BT-199	**Since UBL 2.5**: /CreditNote/cac:CreditNoteLine/cac:Delivery/cac:DeliveryNoteLineReference/cbc:LineID	New BT
BT-196	**Since UBL 2.4**: /CreditNote/cac:CreditNoteLine/cac:Item/cac:ClassifiedTaxCategory/cbc:ItemSupplyTypeCode	New BT

As all later UBL 2.x versions are backward compatible with UBL 2.5, implementations should use the latest backward-compatible UBL 2.x version published by the OASIS UBL Technical Committee [5].
The last column of the mapping table, entitled "Discrepancy and resolution", may include version-related information of the following types in addition to other content:

1. indication that a data syntax element was introduced after UBL 2.1, e.g. "*since UBL 2.5*"
2. indication that the syntax cardinality was subsequently aligned with the semantic cardinality, e.g. "*Cardinality since CII D22B*". The latter does not apply to UBL

4.3 Mismatches and Discrepancies

4.3.1 Semantic alignment

The first step in mapping a semantic model to a syntax is to determine if each element in the semantic model has a corresponding element in the syntax. The corresponding element in the syntax shall have a similar or wider semantic definition with respect to the definition of the semantic model element. The definition of the syntax element may be implied by the name of that element. For example: an element named “VAT Amount” in the semantic model may be mapped to an element named “Tax Amount” in the syntax specification. As VAT is a type of tax, the element “Tax Amount” is a wider concept than VAT Amount. The semantic relation between elements from the semantic model and elements from the syntax specification can be specified using SKOS [7] relation types
The following types of semantic mismatches between individual elements that may occur are listed in Table 3:

Table 3 —Semantic alignment

ID	SOURCE (Semantic)	TARGET (Syntax)	Example	Issue	Resolution
SEM-1	wider	smaller	SOURCE specifies ‘Taxes’, while TARGET specifies ‘VAT’ (SKOS: narrower)	The semantic rules of TARGET may be violated.	1) find another element in TARGET to put the violating instances (those taxes that are not VAT) 2) accept that you are abusing an element in TARGET for something it was not (entirely) designed for. 3) Request to widen semantic definition of TARGET
SEM-2	smaller	wider	SOURCE specifies "VAT", Target specifies "Taxes" (SKOS: broader)	All instances that comply to SOURCE will also comply to TARGET, but some of the semantics are lost: the type of Tax is not specified any more.	Unless other elements are mapped to the wider element as well, specify the narrower meaning in the documentation (VAT instead of Tax).
SEM-3	overlap	overlap	SOURCE specifies Employee (including teachers, staff, researchers –that are on payroll- etc) and TARGET specifies Researcher (can be both enlisted as employee, but also be a student). (SKOS: related)	The semantic rules of TARGET may be violated.	1) accept that you are abusing an element in TARGET for something it was not (entirely) designed for. 2) Request to widen semantic definition of TARGET
SEM-4	match	no match	TARGET is missing any element to specify a person.	It is not possible to put certain information in the TARGET.	1) Use a (more) generic element 2) Request to add an element in TARGET.

4.3.2 Structural alignment

The second step is to review the “structural context” of the information element in the respective syntaxes. The structural context of an element is part of its semantic definition. Electronic messages in the different syntaxes represent data in different levels, groupings and sequences. For example, a VAT Amount element on line level in the model should not be mapped on a VAT Amount element on document level in the syntax specification.
Structural mismatches are listed in Table 4:

Table 4 —Structural alignment

ID	SOURCE (Semantic)	TARGET (Syntax)	Example	Issue	Resolution
STR-1	Hierarchical order one to many	Hierarchical order many to one	Packing of items can be listed as items and then where they are packed or as a list of packs and what items are in each pack.	Yes	Complex mapping. Packs are lifted to higher level and equivalent packs need to be combined.
STR-2	element on higher level	element on lower level	SOURCE specifies element at top level with a single repetition but TARGET is in a class that is also used for other data that requires repetition of the class.	Possibly if higher level cardinalities cause conflicts.	
STR-3	grouping A-B-C	different grouping	SOURCE may define a group of elements such as payment instructions that may be repeated as a group but if those elements are differently grouped in TARGET, that repetition may be problematic.	Possibly.	
STR-4	higher detail	less detail	SOURCE has <name/firstname and name/lastname> and TARGET only has <name>.	Yes.	Agree on a rule to concatenate elements from SOURCE to TARGET
STR-5	less detail	higher detail	TARGET has <name/firstname and name/lastname> and SOURCE only has <name>.	Depends	Agree on a rule (if possible) to split the SOURCE element into several TARGET elements

4.3.3 Cardinality assessment

Cardinality defines whether or not an element shall be used, may be omitted and how many times it might be repeated in a specific context. The cardinality of an element in the syntax shall be the same or less restrictive than the corresponding element in the model. An element that is mandatory in the model may be optional in the syntax specification, but not the other way around. An element that is repeating in the model shall also be repeating in the syntax specification.
The cardinality mismatches that may occur are listed in Table 5:

Table 5 —Alignment of cardinalities

ID	SOURCE	TARGET	Example	Issue	Resolution
CAR-1	optional (0..x)	mandatory (1..x)		If the element is not present, the target rules are violated.	Agree on ‘default value if missing’ (e.g. 0, 1-1-1970, AAA).
CAR-2	mandatory (1..x)	optional (0..x)		None.	Add a rule in the target that the element shall be present.
CAR-3	single (X..1)	multiple (X..n)		None.	Add a rule in the target that the element shall not be repeated.

ID	SOURCE	TARGET	Example	Issue	Resolution
CAR-4	multiple (X..n)	single (X..1)		Repeating elements cannot be handled.	1) If possible, repeat a higher level in the structure 2) In the case of text elements, concatenate the repeating elements
CAR-5	element missing	element mandatory		Yes.	Agree on 'default value if missing' (e.g. 0, 1-1-1970, AAA).

4.3.4 Datatype Alignment

The semantic data types of EN 16931 are listed in Table 6:

Table 6 —Semantic data types

Basic type	Definition
Amount. Type	An amount states a numerical monetary value. The currency of the amount is defined as a separate business term. This EN 16931_ Amount. Type is based on the Amount. Type as defined in ISO 15000-5:2014 Annex A. EN 16931_ Amount. Type is floating up to two fraction digits.
Unit Price Amount. Type	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. This EN 16931_ Unit Price_ Amount. Type is based on the Amount. Type as defined in ISO 15000-5:2014 Annex A.
Quantity. Type	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. This EN 16931_ Quantity. Type is based on the Quantity. Type as defined in ISO 15000-5:2014 Annex A. EN 16931_ Quantity. Type is floating up to five fraction digits.
Percentage. Type	Percentages are given as fractions of a hundred (per cent) e.g. the value 34.78 % in percentage terms is given as 34.78. This EN 16931_ Percentage_ Numeric. Type is based on the Numeric. Type as defined in ISO 15000-5:2014 Annex A. EN 16931_ Percentage_ Numeric.
Identifier. Type	Identifiers (IDs) are keys that are issued by either the sender or recipient of a document or by a third party. For each identifier in the model it is stated whether an identification scheme shall be defined and if so, from what list the identification schemes may be chosen. This EN 16931_ Identifier. Type is based on the Identifier. Type as defined in ISO 15000-5:2014 Annex A. The Scheme identifier Identifies the scheme on which the identifier is based. The use of this attribute is specified for each information element in the semantic model.
Document Reference. Type	Identifiers that were assigned to a document or document line by the Buyer, the Seller or by a third party. This EN 16931_ Document Reference_ Identifier. Type is based on the Identifier. Type as defined in ISO 15000-5:2014 Annex A.
Date. Type	Dates shall be in accordance to the "Calendar date complete representation" as specified by ISO 8601 (see ISO 8601:2004, 5.2.1.1). Calendar dates do not include a specification for the time of the day. This EN 16931_ Date_ Date Time. Type is based on the Date Time. Type as defined in ISO 15000-5:2014 Annex A.
Text. Type	Text is the actual wording of anything written or printed. The language of the textual business terms in the invoice is defined in a separate business term in the model (BT-4 Invoice language code). This EN 16931_ Text. Type is based on the Text. Type as defined in ISO 15000-5:2014 Annex A. Line breaks in the text may be present.
Time Type	Times shall be represented as specified by ISO 8601 (see ISO 8601-1:2019, Date and time — Representations for information interchange — Part 1: Basic rules). The representation shall include timezone information. This EN 16931_ Time_ Date Time. Type is based on the Date Time. Type as defined in ISO 15000-5:2014, Annex B. The content of the Date Time. Format. Text attribute is left to the syntax in which the Time is represented.
Binary Object. Type	Binary objects can be used to describe files which are transmitted together with the Invoice. Attachments shall be transmitted together with the Invoice. There shall be only one way defined per syntax. This EN 16931_ Binary Object. Type is based on the Binary Object. Type as defined in ISO 15000-5:2014 Annex A. EN 16931_ Binary Object. Type 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.

These data types are further defined in EN 16931-1. These definitions are based on the data type definitions in ISO 15000-5. Data types are composites, consisting of a content and zero or more supplementary components. Syntax specifications may deviate from the EN 16931-1 definitions, while being based on the same ISO 15000-5 data types. For example, the set of supplementary components may be different. They also may define different restrictions, such as field lengths.
The issues that may arise at data format level when mapping the model to a syntax are listed in Table 7: