TECHNICAL SPECIFICATION

ISO/TS 20625

First edition
2002-05-01

Electronic data interchange for administration, commerce and transport (EDIFACT) — Rules for generation of XML scheme files (XSD) on the basis of EDI(FACT) implementation guidelines

Échange de données informatisé pour l'administration, le commerce et le transport (EDIFACT) — Règles pour la génération de fichiers de schéma XML (XSD) basés sur les guides de mise en œuvre d'EDI(FACT)

Reference number
ISO/TS 20625:2002(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

In other circumstances, particularly when there is an urgent market requirement for such documents, a technical committee may decide to publish other types of normative document:

— an ISO Publicly Available Specification (ISO/PAS) represents an agreement between technical experts in an ISO working group and is accepted for publication if it is approved by more than 50 % of the members of the parent committee casting a vote;

— an ISO Technical Specification (ISO/TS) represents an agreement between the members of a technical committee and is accepted for publication if it is approved by 2/3 of the members of the committee casting a vote.

An ISO/PAS or ISO/TS is reviewed after three years with a view to deciding whether it should be confirmed for a further three years, revised to become an International Standard, or withdrawn. In the case of a confirmed ISO/PAS or ISO/TS, it is reviewed again after six years at which time it has to be either transposed into an International Standard or withdrawn.

Attention is drawn to the possibility that some of the elements of this Technical Specification may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/TS 20625 was prepared by DIN (as DIN 16557-5) and was adopted, under a special “fast-track procedure”, by Technical Committee ISO/TC 154, *Processes, data elements and documents in commerce, industry and administration*, in parallel with its approval by the ISO member bodies.

Annex A of this Technical Specification is for information only.

Introduction

Traditional EDI standards provide a syntax for the implementation of data content in various business processes through the use of data elements, segments and message types. Initially XML provides simply another syntax, which, if used to re-invent EDI, leads to huge new costs thus preventing any achievement of the initial goal - to get small and medium sized enterprises (SME) involved in electronic business processes.

This standard describes how existing EDI know-how can be applied to the XML syntax. XML users would therefore be able to easily use EDI data from existing applications in a consistent manner.

EDIFACT Message Implementation Guidelines (MIGs) describe the implementation of standardised EDIFACT message types within a business process. Therefore MIGs are the suitable source for the derivation of XML schemas. This standard specifies the process of translation.

Electronic data interchange for administration, commerce and transport (EDIFACT) — Rules for generation of XML scheme files (XSD) on the basis of EDI(FACT) implementation guidelines

1 Scope

This standard describes the rules for the derivation of XML schemas from EDI MIGs providing a sound method of representing semantic facts.

This standard describes how to derive XML from UN/EDIFACT MIGs. In principle, the rules are equally applicable to other EDI standards.

This standard does not apply to DTDs.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this Technical Specification. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Technical Specification are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 8601:2000-12, *Data elements and interchange formats — Information interchange — Representation of dates and times.*

ISO 9735-1:1998-10, *Electronic data interchange for administration, commerce and transport (EDIFACT) – Application level syntax rules (Syntax version number 4, Syntax release number: 1) – Part 1: Syntax rules common to all parts.*

3 Terms, symbols and abbreviations

For the purpose of this standard the following terms, symbols and abbreviations apply.

3.1
BSR
Basic Semantics Register

3.2
BSU
Basic Semantic Unit

3.3
DTD
Document Type Definition

3.4
EDI
Electronic Data Interchange

3.5
EDIFACT
Electronic Data Interchange for Administration, Commerce and Transport

3.6
ELEMENT
Syntactic building block containing data and/or attributes.

3.7
HTML
Hyper Text Mark-up Language

3.8
MIG
Message Implementation Guideline

3.9
NAME
A name in the context of XML starts with a letter or a permitted special character followed by letters, numbers, hyphens, underlines, colons or points. All those are known as name tag. Names starting with "xml" or with a character string which fits to (('X'|'x') ('M'|'m') ('L'|'l') are reserved for XML standardisation.

3.10
SGML
Standard Generalised Mark-up Language

3.11
Tag
Formatting instruction or semantic markup

3.12
Template
Predefined reference pattern compared with the complete entity to be recognised, or one of its parts.

3.13
XLL
Extensible Link Language

3.14
XML
Extensible Mark-up Language

3.15
XSD
Extensible Schema Definition

3.16
XSL
Extensible Stylesheet Language

3.17
W3C
World Wide Web Consortium

4 Typical contents of Message Implementation Guidelines

4.1 Level: MIG

a) Identification of MIG

b) Identification of the supporting EDIFACT directory

c) Identification of the message type and, if necessary, the industry subsets

d) Additional text

4.2 Level: Message Type

a) Structure of the message type (segments and segment groups) and indication of their portions used

b) Status (standard versus application) of the segments and segment groups in use

c) Context related names and descriptions of the segments and segment groups

d) Examples

e) Dependencies between segments and segment groups

f) Additional text, comments on message type level

4.3 Level: Segments and Composite Data Elements

a) Structure of the segments and composite data elements and indication of their portions used

b) Status (standard versus application) of the data elements and composite data elements

c) Dependencies between data elements and composite data elements within a segment and within the message type

d) Context related names and descriptions

e) Examples

f) Additional text, comments

4.4 Level: Data element

a) Characteristics of EDI data elements (type, length) and their usage restrictions based upon MIG and context related implementation

b) Context related names and descriptions of data elements and, if necessary, unique tags and descriptions, e. g. derived from data repositories such as ISO-BSR (see ISO/TS 16668).

c) Examples

d) Additional text, comments

e) Allowed values

f) Constants

g) Explicitly given EDIFACT codes or ISO/UN code lists

h) Explicitly given user defined codes

i) Implicitly given EDIFACT codes or ISO/UN code lists

j) Implicitly given user defined or other codes not listed within the EDIFACT code directory

k) Rules to which data element values shall fit

l) Mapping to fields within applications and flat files, respectively

5 Requirements of derivation rules for schemas

a) The MIG technical information as listed in section 4 shall be incorporated into schemas as necessary.

b) The structure of the underlying MIG must be comprehensible (both the XML and traditional EDI guides shall be compatible in structure).

c) The resulting XML messages should be as lean as possible.

d) One of the different variants by which semantic facts can be represented in XML is specified by this standard as being mandatory.

e) The developer of a MIG decides which data is important and which structures are meaningful for his application. By this, he decides which structure elements shall be incorporated into the schema.

6 Rules for the generation of XML schemas derived from EDI MIGs

NOTE The namespace ‘din’ in the examples of this section is for example purposes only and can either be omitted or any other suitable namespace can be used.

6.1 Rule 1: Tag naming

6.1.1 Variant 1

The names of the XML structure will be derived from the EDI tags. They will be given a prefix depending on the structure level (segment group, segment, composite data element or data element):

„M_“+ message type + [suffix]	Example: M_ORDERS
„G_“+ segment group + [suffix]	Example: G_SG36 or G_LIN_ALC
„S_“+ segment + [suffix]	Example: S_LIN
„C_“+ composite data element + [suffix]	Example: C_C082_2
„D_“+ data element + [suffix]	Example: D_3035 or D_3035_10

The suffix is optional and can be generated based upon various semantic understanding of EDI elements.

If the XML schema file is being generated from an EDIFACT MIG only prefix "D_" would be necessary. However, as the other prefixes have to be used by those EDI standards which identify composite data elements and data elements by using numeric tags, they are mandatory.

The second notation of segment group tags can be used when the based EDI standard that is being converted from provides no explicite segment groups or whenever the notation of the relevant trigger segments is prefered. In this case the nesting of the segment groups has to be given as sequence of their trigger segments.

The W3C XML recommendation requires ”self-explanatory tags“. EDI[FACT] tags fulfil this condition better than tags in natural language, because they represent an established Lingua Franca for EDI specialists.

Example:

```
<xsd:element name ="M_ORDERS">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="din:D_1004"/>
        <xsd:element ref="din:D_2380"/>
        <xsd:element ref="din:D_2380_2"/>
        <xsd:element ref="din:G_SG2"/>
        <xsd:element ref="din:G_SG2_2"/>
        <xsd:element ref="din:D_6345" minOccurs="0" maxOccurs="5"/>
        <xsd:element ref="din:G_SG25" minOccurs="1" maxOccurs="10"/>
        <xsd:element ref="din:D_5004_2"/>
      </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```


6.1.2 Variant 2

From suitable comments ”speaking" tags can be generated if desired. If using "speaking" tags the EDI origin of the corresponding element shall be documented by an appropriate attribute value (see also section 6.9) or with other means of documentation.

Example:

```
  <xsd:element name ="M_ORDERS">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="din:Order_number"/>
        <xsd:element ref="din:Order_date"/>
        <xsd:element ref="din:Delivery_date"/>
        <xsd:element ref="din:Buyer"/>
        <xsd:element ref="din:Seller"/>
        <xsd:element ref="din:Currency" minOccurs="0" maxOccurs="5"/>
        <xsd:element ref="din:Line_item_details" minOccurs="1"
         maxOccurs="10"/>
        <xsd:element ref="din:Total_order_value"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
...
  <xsd:element name ="Name">
    <xsd:complexType>
      <xsd:simpleContent>
        <xsd:extension base ="string1..10">
          <xsd:attribute name="EDIPath" type="xsd:string"
fixed="ORDERS.SG2.NAD.C080.3036(0120:040:01)"/>
```

```
<!-- The attribute EDIPath contains the reference to the original EDI
standard -->
        </xsd:extension>
      </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

6.2 Rule 2: Structure

6.2.1 The same EDI tags or names will produce aggregate elements (see also rule 6.10).

6.2.2 If a differenciation between different semantic occurencies of the same data container is desired, different tags or names have to be assigned either by adding a suffix to the EDI tag or by using different names.

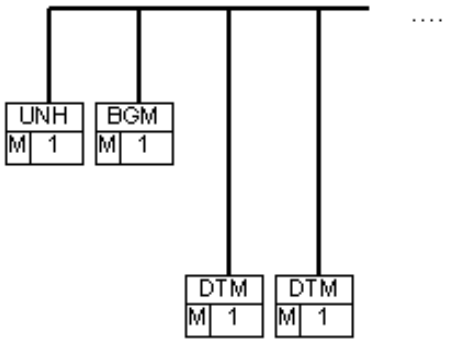
6.2.3 The schema may contain further ‘stapling’ elements for groups of messages or the interchange itself (comparable with UN/EDIFACT UNG-UNE and UNB-UNZ).

6.2.4 Any use of an EDI data container (message type, segment group, segment etc.) can be shown as an independent XML element. The existing EDI structure is the source of the XML structure. Hence the XML schema must have a structure compatible to the EDI MIG. The set of the generated XML elements is less than or equal to the set of EDI elements.

NOTE The manner in which the author has written a MIG must have satisfied the needs of the respective business process. Thus the schema must be structured accordingly. If for example the MIG contains "document date“ and "requested delivery date“ in two separate occurrences of the DTM segment separate XML elements will be generated accordingly. If those two have been documented within the same occurrence of the DTM segment, only one XML element will be generated.

Examples for 6.2.1 and 6.2.2:

Variant 1:
The guide contains two DTM segments (see figure 1).




```
DTM (#1), Status M, Occurrence 1, Qualifier in DE 2005: 4, Name: Order date
DTM (#2), Status M, Occurrence 1, Qualifier in DE 2005: 2, Name: Requested
delivery date
```

Figure 1 — Message diagram of a guide containing two DTM segments

The default translation into an XML schema according 6.2.1 is as follows:

```
<xsd:element name ="S_DTM">
  <xsd:complexType>
```

```
        <xsd:sequence>
          <xsd:element ref="din:D_2005"/>
          <xsd:element ref="din:D_2380"/>
        </xsd:sequence>
      </xsd:complexType>
    <xsd:element name ="D_2005" type ="din:D_2005">
      <xsd:annotation>
        <xsd:documentation>Type of date</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name ="D_2380" type ="xsd:decimal">
      <xsd:annotation>
        <xsd:documentation>Date/Time/Period</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
```

NOTE Element D_2005 is an enumeration type and contains the two possible values '2' and '4'.

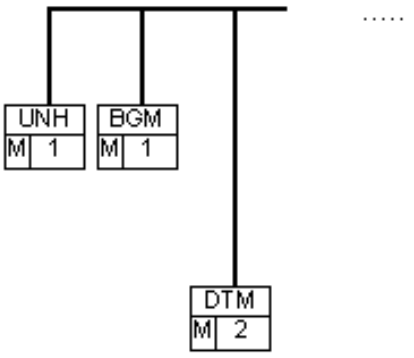
Alternatively, application of rule 6.2.2 would result

```
  <xsd:element name ="D_2380" type ="xsd:decimal">
    <xsd:annotation>
      <xsd:documentation>Order date</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name ="D_2380_2" type ="xsd:decimal">
    <xsd:annotation>
      <xsd:documentation>Delivery date</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
```


or

```
  <xsd:element name ="Order_date" type ="xsd:decimal"/>
  <xsd:element name ="Delivery_date" type ="xsd:decimal"/>
```


Variant 2:
Guide implicitly documenting dates using only one DTM segment (see figure 2).



DTM, Status M, Occurrence 2, Qualifier in DE 2005: 2 and 4

Figure 2 — Message diagram of a guide containing only one DTM segment

The translation into an XML schema is analogue to the default example according to rule 6.2.1 as follows:

```
    <xsd:element name ="S_DTM">
      <xsd:complexType>
```

```
      <xsd:sequence>
        <xsd:element ref="din:D_2005"/>
        <xsd:element ref="din:D_2380"/>
      </xsd:sequence>
    </xsd:complexType>
  <xsd:element name ="D_2005" type ="din:D_2005">
    <xsd:annotation>
      <xsd:documentation>Type of date</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name ="D_2380" type ="xsd:decimal">
    <xsd:annotation>
      <xsd:documentation>Order date</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
```

Example for 6.2.3:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name ="S_UNB">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="D_0004" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="D_0010" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="D_0017" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="D_0020" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="M_ORDERS" minOccurs="1" maxOccurs="unbounded"/>
      </xsd:sequence>
      ...
```


6.3 Rule 3: Structure optimisation

If flat XML structures are of primary interest, the application of the following rules will give an optimised result. However, for integration in existing systems, one should bear in mind the minimum data structure requirements established by the EDI system in use rather than the pure syntax requirements.

6.3.1 An EDIFACT segment that contains more than one data element with business data has actually a summarising function. If the segment only contains one data element with business data there is no summarising function on the segment level. Therefore in transformation towards XSD schema this segment level can disappear.

6.3.2 Elements of the primary standard not being used in the MIG will be omitted.

6.3.3 Constant qualifiers or constant codes are not transferred into the XML structure (for a definite data element only one code had been documented in the MIG). The corresponding data elements should not be transferred into XML.

Example:

Derived from:

```
<xsd:element name ="S_DTM">
 <xsd:complexType>
   <xsd:sequence>
     <xsd:element ref="din:C_C507"/>
   </xsd:sequence>
 </xsd:complexType>
</xsd:element>
```

```
 <xsd:element name ="C_C507">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="din:D_2005"/>
      <xsd:element ref="din:D_2380"/>
      <xsd:element ref="din:D_2379"/>
    </xsd:sequence>
  </xsd:complexType>
 </xsd:element>
<xsd:element name ="D_2379" fixed ="102"/>
<xsd:element name ="D_2005" fixed ="4"/>
<xsd:element name ="D_2380" type ="xsd:decimal">
  <xsd:annotation>
    <xsd:documentation>Order date</xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

This rule generates:

```
<xsd:element name ="D_2380" type ="xsd:decimal">
    <xsd:annotation>
      <xsd:documentation>Order date</xsd:documentation>
    </xsd:annotation>
</xsd:element>
```

The levels of segments and composite data elements are not required as they contain constant qualifiers only. Therefore they are omitted.

6.4 Rule 4: Status

EDI status and application status within the MIG will be summarised in the XML status. The more restrictive status will be kept.

The status “mandatory“ will be represented by a minimum repeating factor of "1", the status ”conditional“ by a minimum repeating factor of "0". The status is given by the attribute **minOccurs.**

Examples:

Conditional:

Segment group	`<xsd:element ref="din:G_SG7" minOccurs="0" maxOccurs="5"/>`
Segment	`<xsd:element ref="din:S_IMD" minOccurs="0" maxOccurs="1"/>`
Composite data element	`<xsd:element ref="din:C_C059" minOccurs="0" maxOccurs="1"/>`
Data element	`<xsd:element ref="din:D_4022" minOccurs="0" maxOccurs="1"/>`

Mandatory:

Segment group	`<xsd:element ref="din:G_LIN" minOccurs="1" maxOccurs="10"/>`
Segment	`<xsd:element ref="din:S_LIN" minOccurs="1" maxOccurs="1"/>`
Composite data element	`<xsd:element ref="din:C_C516" minOccurs="1" maxOccurs="1"/>`
Data element	`<xsd:element ref="din:D_0065" minOccurs="1" maxOccurs="1"/>`

6.5 Rule 5: Maximum occurrences

The number of occurrences of the MIG generates the XML occurrence. The value will be provided using the XSD attribute **maxOccurs**.

Example:

Segment group `<xsd:element ref="din:G_SG25" minOccurs="1" maxOccurs="10"/>`
Segment `<xsd:element ref="din:S_LIN" minOccurs="1" maxOccurs="1"/>`

From EDIFACT syntax version 4 (ISO 9735-1) and the implementation of appropriate directories the occurrence rule is applicable with composite data elements and data elements.

6.6 Rule 6: Data element formats

6.6.1 The representation ”an“ and ”a“ become „string“, ”n“ becomes „decimal“. For the lengths of alphanumeric and numeric data elements as defined within the MIG appropriate simpleTypes will be generated.

6.6.2 Date formats may be translated into XML data types "date", "timeInstant" and "time". In this case a format conversion is required. The representation of these formats within XML is:

date : `1999-05-31 (according to ISO 8601)`
time : `13:20:00`
timeInstant: `1999-05-31T13:20:00`

Example:


```
<xsd:simpleType name="string1..70">
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="70"/>
    </xsd:restriction>
  </xsd:simpleType>
```

6.7 Rule 7: Code lists and user defined codes

6.7.1 Coded data elements will be defined as complexType. If for a data element only specific codes are documented within the MIG, only these codes are allowed for the application. Thus only these codes are transferred into the XML structure.

6.7.2 If the MIG does not provide codes for a data element, the whole available code list is allowed. This whole code list will be transferred into the XML structure.

6.7.3 Repeatedly used code lists may be provided by using external files.

6.7.4 The code names will optionally be stored as annotation with the code.

6.7.5 According to rule 3 (see section 6.3) constant qualifiers or constant codes will not be transferred into the XML structure (for a specific data element only one code had been documented within the MIG). The respective data elements need not be provided within the XML structure. However, if the usage of a data element is explicitly required, it has to be included within the XML structure (e. g. a currency using data element 6345 in segment MOA).

Examples:

(1)

```
<xsd:element name ="D_6345" type ="D_6345"/>
<xsd:simpleType name="D_6345">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="DEM">
        <xsd:annotation>
          <xsd:documentation>Deutsche Mark</xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
      <xsd:enumeration value="EUR">
        <xsd:annotation>
          <xsd:documentation>Euro</xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
      <xsd:enumeration value="GBP">
        <xsd:annotation>
          <xsd:documentation>Pfund Sterling</xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
    </xsd:restriction>
  </xsd:simpleType>
```

(2)

```
<xsd:simpleType name="D_6347">
 <xsd:restriction base="xsd:string">
    <xsd:enumeration value="1"/>
    <xsd:enumeration value="2"/>
    <xsd:enumeration value="3"/>
    <xsd:enumeration value="4"/>
    <xsd:enumeration value="5"/>
    <xsd:enumeration value="6"/>
    <xsd:enumeration value="7"/>
   ... etc. listing of the complete code list
</xsd:simpleType>
```


(3)

```
<?xml version="1.0"?>
<xsd:schema targetNamespace="http://www.din.de/example/orders"
             xmlns:din="http://www.din.de/example/orders"
             xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
<include schemaLocation="CL_6411.xsd"/>
<xsd:element name ="D_6411" type ="din:CL_6411">
    <xsd:annotation>
      <xsd:documentation>Measure unit</xsd:documentation>
    </xsd:annotation>
</xsd:element>
...
```

External file with codes:

```
<?xml version="1.0"?>
<xsd:schema targetNamespace="http://www.din.de/ example/orders"
             xmlns:din="http://www.din.de/ example/orders"
             xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
<xsd:simpleType name="CL_6411">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="ACR"/>
      <xsd:enumeration value="AMH"/>
   ......
```