TECHNICAL SPECIFICATION

ISO/TS 21377

First edition
2023-02

Exchange formats for the audit data collection standard: XML and JSON

Reference number
ISO/TS 21377:2023(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 295, *Audit data services*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

Accounting and enterprise resource planning (ERP) software packages are widely used in businesses and by various government organizations to manage and track business processes, post transactions and produce financial reports. Because of the nature of the information contained within the ERP systems, the data are also leveraged by internal and external auditors to assess the business controls, processes and financial reporting. There are numerous ERP packages that are used by businesses and government organizations, which can vary greatly in design (e.g. interfaces, data content, data formats, operational reports, management reports, financial reports). These and other design differences present challenges in the collection of data for auditing supervision management purposes.

This document concerns the specification of technical exchange formats as output file formats for the functional content defined in ISO 21378.

Exchange formats for the audit data collection standard: XML and JSON

1 Scope

ISO 21378, the audit data collection standard (ADCS), defines the functional requirements for exchanging audit data in flat file format. This document concerns the specification of technical exchange formats in extensible markup language (XML), JavaScript object notation (JSON) and comma-separated values (CSV) as output file formats for the functional content defined in ISO 21378.

This document also contains the following schemas and sample files for tables in ISO 21378.

— XML schema;

— XML sample file;

— JSON schema;

— JSON sample file.

To keep the three exchange formats (XML, JSON and CSV) consistent, this document also specifies how to use the technical solution in the CSV format.

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.

ISO 21378:2019, *Audit data collection*

ISO 8601-1, *Date and time — Representations for information interchange — Part 1: Basic rules*

ISO/IEC 21778, *Information technology — The JSON data interchange syntax*

W3C *Extensible Markup Language (XML) 1.0 (Fifth Edition).* Available at https://www.w3.org/TR/2008/REC-xml-20081126/

W3C *XML Schema Part 1: Structures Second Edition*. Available at https://www.w3.org/TR/2004/REC-xmlschema-1-20041028/

W3C *XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes*. Available at https://www.w3.org/TR/2012/REC-xmlschema11-2-20120405/

JSON Schema: A Media Type for Describing JSON Documents. Available at https://json-schema.org/draft/2020-12/json-schema-core.html

3 Terms and definitions

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

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
data
reinterpretable representation of information in a formalized manner suitable fo r communication, interpretation or processing

[SOURCE: ISO/IEC 11179-1:2015, 3.2.6, modified — Notes to entry have been deleted.]

3.2
data element
basic unit of identifiable and definable *data* (3.1)

[SOURCE: ISO 2146:2010, 3.4, modified — The admitted term "element" has been deleted.]

3.3
data file
collection of *data* (3.1) records having a homogeneous structure

[SOURCE: ISO 21378:2019, 3.4]

3.4
data structure
framework comprising a number of *data elements* (3.2) in a prescribed form

[SOURCE: ISO 21007-1:2005, 2.16, modified — The word "element" has been deleted from the term.]

3.5
syntax
set of rules, principles and processes that govern the *data structure* (3.4)

3.6
data model
graphical and/or lexical representation of *data* (3.1), specifying their properties, structure, and inter-relationships

[SOURCE: ISO/IEC 11179-1:2015, 3.2.7]

3.7
entity
group of *data elements* (3.2) describing an object

Note 1 to entry: It is equivalent to "object class" in ISO/IEC 11179-1.

3.8
relation
relationship between two *entities* (3.7)

3.9
attribute
data element (3.2) describing an object

Note 1 to entry: It is equivalent to "property" in ISO/IEC 11179-1.

3.10
domain
set of properties to define the value space of *attributes* (3.9)

Note 1 to entry: A domain contains *code lists* (3.11) and *code values* (3.12). It is equivalent to "representation" in ISO/IEC 11179-1.

3.11
code list
standardized list of *code values* (3.12) with a common scope

3.12
code value
one value from a *code list* (3.11)

4 Exchange formats

4.1 General

The data format is the carrier of data exchange between auditor and auditee. So, it is necessary to make an agreement on data format between the two sides in data exchange. There are multiple options for output data formats.

In the case of ADCS, three exchange formats are specified:

— XML data files defined by XML schema specification (W3C);

— JSON data files defined by JSON schema specification (json-schema.org);

— CSV data files.

These three formats shall contain the same functional ADCS content specification in accordance with ISO 21378 but differ on the technical level only.

This means that these exchange formats are convertible from one to the other. For instance, it is possible to convert XML data files to CSV data files or CSV data files to JSON data files.

To generate the XML and JSON schemas in a consistent way, a data model is set up from ISO 21378 according to ISO/IEC 11179-1. This data model is used to create consistent exchange format specifications.

Each ADCS table is modelled into a "Parent Entity" and reusable data groups within a table are modelled into "Child Entities", for example, "Physical Address", "Billing Address", "Tax", "Created", "Modified", "Posted", completely in line with ISO 21378. This way guarantees that all reusable groups that occur in multiple tables are defined in the same way.

As the data model is not in scope of this document, it is only documented for reference purposes in Annex D.

The various technical specifications are explained in this document; and the additional packaging and communication agreements are defined.

4.2 XML

4.2.1 General

This subclause describes how to exchange the ADCS tables using XML data files.

XML data files shall be created in accordance with the syntax specifications written in W3C Extensible Markup Language (XML) 1.0 (Fifth Edition) and also in accordance with the functional requirements written in ISO 21378.

4.2.2 Technical requirements

4.2.2.1 General

Only one ADCS table, with a number of lines, shall be sent in one XML data file.

The filename shall comply with the rules described in 4.6.5.

Each ADCS table has its own XML schema to define its XML file. When there are 71 tables, there are 71 XML schemas.

Such an XML data file contains (schematic):

```
<?xml version='1.0' encoding='UTF-8'?>
<root-tag
    xsi:schemaLocation='name-space xml-schema-name.xsd'
    xmlns='name-space'
    xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'>
    <table-line-tag>
        <data-element-tag>value</data-element-tag>
        <group-tag>
            <data-element-tag>value</data-element-tag>
        </group-tag>
    </table-line-tag>
</root-tag>
```

Italicized text shall be replaced by the definitions in this document.

4.2.2.2 Root tag

The root tag equals "Adc" followed by the ADCS table name like: “AdcApAdjustments”.

4.2.2.3 Target name space definition

The target name space equals "http://schemas.iso.org/AdcsML/Messages/" followed by the message name and message version, like "http://schemas.iso.org/AdcsML/Messages/AdcApAdjustments-v1".

4.2.2.4 XML tags

Table-line-tag, group-tag and data-element-tag are XML tags which are defined in the related XML schema.

XML tags are in the first step derived from the full data element names, and in the second step shortened in a consistent manner, according to an abbreviation list (see Annex A).

Because auditors and auditees usually exchange bulk files according to ADCS, abbreviated tags can reduce the file sizes by 30 % to 40 %.

Full data element names and abbreviated XML tags are together specified in ADCS to XML mapping tables (see explanation and example of mapping tables in 4.5).

In XML schemas the full data element names are documented as annotations.

This means that programmers can have easy access to the full data element names, either from the mapping tables or from the XML schemas.

4.2.2.5 Optional fields

Optional fields and optional groups of fields can be omitted from the XML data file, only if these data fields are not available in the source system that delivers the audit data.

4.2.2.6 Repeating groups

Repeating groups like “Tax” can occur up to the maximum number that is defined in the XML schema.

4.2.2.7 Special characters to be escaped

The following special characters shall be escaped in XML data files as follows:

< (less than) to be replaced with <

> (greater than) to be replaced with >

& (ampersand) to be replaced with &

‘ (apostrophe) to be replaced with '

“ (quotation mark) to be replaced with "

4.2.3 XML schema

4.2.3.1 General

Each ADCS file is technically specified by an XML schema and can also be validated by using that schema.

XML schema files shall be created in accordance with the syntax specifications written in W3C Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C XML Schema Part 1: Structures Second Edition, W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes and also in accordance with the functional requirements written in ISO 21378.

Table 1 shows ISO 21378 representation formats and the equivalents used in XML schema.

Table 1 — Representation specification in XML schema

ISO 21378 representation	XML schema representation
%ns	<xsd:restriction base="xsd:string"> <xsd:maxLength value="n"/> </xsd:restriction>
%nc	<xsd:restriction base="xsd:string"> <xsd:length value="n"/> </xsd:restriction>
%m.nf	<xsd:restriction base="xsd:decimal"> <xsd:totalDigits value="m"/> <xsd:fractionDigits value="n"/> </xsd:restriction>
%nd	<xsd:restriction base="xsd:integer"> <xsd:totalDigits value="n"/> </xsd:restriction>

Table 2 shows ISO 21378 datatypes and the equivalents used in XML schema.

Table 2 — Datatypes in XML schema

ISO 21378 datatype	XML schema datatype
Date	<xsd:restriction base="xsd:date">
Time	<xsd:restriction base="xsd:time">
String	<xsd:restriction base="xsd:string">
Decimal	<xsd:restriction base="xsd:decimal">
Integer	<xsd:restriction base="xsd:integer">
Boolean	<xsd:restriction base="xsd:boolean"/>

All schema files are bundled in one official ISO delivery package. Another development package is also available.

a) Official ISO delivery package: This package contains the schemas established in this document and ISO 21378 for exchanging the audit data.

b) Development package: This package contains the schemas that are used to develop new versions of the existing schemas and for the development of extensions.

The electronic version of these XML schemas can be downloaded from the URL which can be found in Annex E.

4.2.3.2 Official ISO delivery package

All schema files in this package have one and the same version number, which can be found in the schema file itself.

The first delivery of the package has version number 1. After a certain period, if there are sufficient reasons for this, a new package of schemas will be delivered with a successor version number, for example, version number 2.

This package contains:

a) All ADCS table schemas. The name of the xml schema file equals "Adc" followed by the ADCS table name and then followed by ".xsd", such as "AdcApAdjustments.xsd".

b) One data types schema named "Adc_DataTypes.xsd". This schema is imported into all ADCS table schemas.

A list of these XML schemas can be found in Annex B.

EXAMPLE 1 Part of "AdcApAdjustments.xsd":

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- XML Definition of message hierarchy                                    -->
<!-- Message:        ADC AP Adjustments v1.0                                -->
<!-- Generated at:   25-05-2022 16:53:41                                    -->
<!-- Organisation:   This schema is part of ISO 21377 Technical Specification -->

<xsd:schema     xmlns:str="http://schemas.iso.org/AdcsML/Messages/AdcApAdjustments-v1"
                xmlns:cdt="http://schemas.iso.org/AdcsML/Adc_DataTypes-v1"
                xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                targetNamespace="http://schemas.iso.org/AdcsML/Messages/
AdcApAdjustments-v1"
                elementFormDefault="qualified" attributeFormDefault="unqualified"
version="1.0">

   <xsd:import namespace="http://schemas.iso.org/AdcsML/Adc_DataTypes-v1"
schemaLocation="Adc_DataTypes.xsd"/>
   <xsd:element name="AdcApAdjustments">
     <xsd:annotation>
        <xsd:documentation>MESSAGE</xsd:documentation>
     </xsd:annotation>
     <xsd:complexType>
        <xsd:sequence>
           <xsd:element name="ApAdj" minOccurs="0" maxOccurs="unbounded">
              <xsd:annotation>
                <xsd:documentation>AP ADJUSTMENT</xsd:documentation>
              </xsd:annotation>
              <xsd:complexType>
                <xsd:sequence>

                   <xsd:element name="AdjId" type="cdt:TypId100" minOccurs="1"
maxOccurs="1">
                     <xsd:annotation>
                        <xsd:documentation>Adjustment ID</xsd:documentation>
                     </xsd:annotation>
```



```
                    </xsd:element>

                    <xsd:element name="AdjNr" type="cdt:TypText100" minOccurs="0"
maxOccurs="1">
                       <xsd:annotation>
                          <xsd:documentation>Adjustment Number</xsd:documentation>
                       </xsd:annotation>
                    </xsd:element>

                 </xsd:sequence>
              </xsd:complexType>
           </xsd:element>
        </xsd:sequence>
     </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

EXAMPLE 2 Part of Adc_DataTypes.xsd:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- XML Definition of elementary types                                    -->
<!-- Datamodel:     Audit Data Collection Datamodel 1, version HEAD        -->
<!-- Generated at:  25-05-2022 16:53:41                                    -->
<!-- Organisation:  This schema is part of ISO 21377 Technical Specification   -->

<xsd:schema xmlns:cdt="http://schemas.iso.org/AdcsML/Adc_DataTypes-v1"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://schemas.iso.org/AdcsML/Adc_DataTypes-v1"
            elementFormDefault="qualified"attributeFormDefault="unqualified"
version="1.0">

<!-- IDENTIFIER 100: FORMAT: AN..100 -->
<xsd:simpleType name="TypId100">
   <xsd:restriction base="xsd:string">
      <xsd:maxLength value="100"/>
      </xsd:restriction>
</xsd:simpleType>

<!-- TEXT 100: FORMAT: AN..100 -->
<xsd:simpleType name="TypText100">
   <xsd:restriction base="xsd:string">
         <xsd:maxLength value="100"/>
      </xsd:restriction>
   </xsd:simpleType>

</xsd:schema>
```

4.2.3.3 Development package

This package contains the schemas that are used to develop new versions of the existing schemas and for the development of extensions.

In this package each schema has its own version number, which is included in the version attribute and which is added to the name of the XML schema file. In the first delivery of this package, all schemas have version 1.0 like “AdcApAdjustments-v1.0.xsd”. When a schema changes, the minor version number is incremented each time (e.g. version number changes from 1.0 to 1.1).

In this package, all schemas contain their own data type definitions, so that they can be further developed independently and there is no common data type schema.

After a certain period of time, if there are sufficient reasons to do this, a new official ISO delivery package will be delivered with the next successor major version number, for example, version number 2 (see 4.2.3.2); and the new common data type schema (Adc_DataTypes.xsd) will be delivered.

Also a new development package will be delivered, in which all schemas will have version 2.0.

EXAMPLE Part of “AdcApAdjustments-v1.1.xsd”:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- XML Definition of message hierarchy                                    -->
<!-- Message:  ADC AP Adjustments v1.1                                      -->
<!-- Generated at:    25-05-2022 16:53:41                                   -->
<!-- Organisation:  This schema is part of ISO 21377 Technical Specification     -->

<xsd:schema xmlns:str="http://schemas.iso.org/AdcsML/Messages/AdcApAdjustments-v1"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://schemas.iso.org/AdcsML/Messages/AdcApAdjustments-v1"
            elementFormDefault="qualified"attributeFormDefault="unqualified"
version="1.1">

  <xsd:element name="AdcApAdjustments">
    <xsd:annotation>
      <xsd:documentation>MESSAGE</xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="ApAdj" minOccurs="0" maxOccurs="unbounded">
          <xsd:annotation>
            <xsd:documentation>AP ADJUSTMENT</xsd:documentation>
          </xsd:annotation>
          <xsd:complexType>
            <xsd:sequence>

              <xsd:element name="AdjId" type="str:TypId100" minOccurs="1" maxOccurs="1">
                <xsd:annotation>
                  <xsd:documentation>Adjustment ID</xsd:documentation>
                </xsd:annotation>
              </xsd:element>

              <xsd:element name="AdjNr" type="str:TypText100" minOccurs="0" maxOccurs="1">
                <xsd:annotation>
                  <xsd:documentation>Adjustment Number</xsd:documentation>
                </xsd:annotation>
              </xsd:element>

            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:simpleType name="TypId100">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="TypText100">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>

</xsd:schema>
```

4.2.4 XML sample

The following example shows one XML data file that contains only one record of an ADCS table:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- XML Sample Message of message hierarchy                                -->
<!-- Message:        ADC AP Adjustments v1.0 Rev. 11                        -->
<!-- Generated at:   09-05-2022 13:22:56                                    -->
<!-- Organisation:   This sample is part of ISO 21377 Technical Specification  -->

<AdcApAdjustments xsi:schemaLocation="http://schemas.iso.org/AdcsML/Messages/
AdcApAdjustments-v1../AdcApAdjustments-v1.1.xsd"
    xmlns="http://schemas.iso.org/AdcsML/Messages/AdcApAdjustments-v1"
```

```
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ApAdj>
    <AdjId>AD1234</AdjId>
    <AdjNr>123456789</AdjNr>
    <AdjTypNm>Finance charge.</AdjTypNm>
    <AdjDocNr>123456789</AdjDocNr>
    <InvoiId>I001</InvoiId>
    <JrnId>JRN1</JrnId>
    <FiscYr>2020</FiscYr>
    <AcntingPer>M1</AcntingPer>
    <AdjDt>2020-02-01</AdjDt>
    <SuplAcntId>S00001</SuplAcntId>
    <GlDbAcntNr>ACC123456789</GlDbAcntNr>
    <GlCrAcntNr>ACC123456789</GlCrAcntNr>
    <AdjAmt>
      <FuncAmt>125.1255</FuncAmt>
      <FuncCurCd>USD</FuncCurCd>
      <TrAmt>125.1255</TrAmt>
      <TrCurCd>USD</TrCurCd>
      <RprtAmt>125.1255</RprtAmt>
      <RprtCurCd>USD</RprtCurCd>
      <LocAmt>125.1255</LocAmt>
      <LocCurCd>USD</LocCurCd>
    </AdjAmt>
    <Tax>
      <TaxTypCd>TAX01</TaxTypCd>
      <TaxLocAmt>26.2763</TaxLocAmt>
      <GlTaxDbAcntNr>ACC123456789</GlTaxDbAcntNr>
      <GlTaxCrAcntNr>ACC123456789</GlTaxCrAcntNr>
    </Tax>
    <Tax>
      <TaxTypCd>TAX02</TaxTypCd>
      <TaxLocAmt>10.0000</TaxLocAmt>
      <GlTaxDbAcntNr>ACC123456789</GlTaxDbAcntNr>
      <GlTaxCrAcntNr>ACC123456789</GlTaxCrAcntNr>
   </Tax>
    <Crea>
      <UserId>JD</UserId>
      <Dt>2020-03-02</Dt>
      <Tm>09:31:00</Tm>
    </Crea>
    <Aprv>
      <UserId>JD</UserId>
      <Dt>2020-03-02</Dt>
      <Tm>09:31:00</Tm>
    </Aprv>
    <LstMdf>
      <UserId>JD</UserId>
      <Dt>2020-03-02</Dt>
      <Tm>09:31:00</Tm>
    </LstMdf>
    <BusSg>
      <SqNr>3</SqNr>
      <BusSgCd>BS001</BusSgCd>
    </BusSg>
  </ApAdj>
</AdcApAdjustments>
```


4.3 JSON

4.3.1 General

This subclause describes how to exchange the ADCS tables using JSON data files.

JSON data files shall be created in accordance with the syntax specifications in ISO/IEC 21778 and also in accordance with the functional requirements in ISO 21378.

4.3.2 Technical requirements

4.3.2.1 General

Only one ADCS table, with a number of lines, shall be sent in one JSON data file.

The filename shall comply with the rules described in 4.6.5.

Each ADCS table has its own JSON schema to define its JSON file. So, when there are 71 tables, there are 71 JSON schemas.

JSON uses the same tags as XML.

Such a JSON data file contains (schematic):

```
{
    "table-line-tag": [{
        "data-element-tag": "value",
        "group-tag": {
            "data-element-tag": "value"
        }
    }]
}
```

Italicized text shall be replaced by the definitions in this document.

4.3.2.2 $id

JSON data files do not contain a namespace definition and root tag, in contrast to XML data files. "$id" is the first data element in each JSON data file. The value of $id includes the ADCS table name (for example, AdcApAdjustments) and main version number (for example -v1).

Example of $id:

```
"$id": "http://schemas.iso.org/AdcsML/Messages/AdcApAdjustments_jsd6-v1"
```

4.3.2.3 JSON tags

A JSON data file contains the table-line-tag like "ApAdj" only one time with a number of the table lines.

Table-line-tag, group-tag and data-element-tag are JSON tags which are defined in the related JSON schema.

JSON uses the XML tags. XML tags are in the first step derived from the field names, and in the second step shortened in a consistent manner, according to an abbreviation list (see Annex A).

Because auditors and auditees usually exchange bulk files according to ADCS, abbreviated tags can reduce the file sizes by 30 % to 40 %.

Full data element names and abbreviated XML tags are together specified in ADCS to XML mapping tables (see explanation and example of mapping tables in 4.5).

This means that programmers can have easy access to the full data element names from the mapping tables.

4.3.2.4 Optional fields

Optional fields and optional groups of fields can be omitted from the JSON data file, only if these data elements are not available in the source system that delivers the audit data.

4.3.2.5 Repeating groups

Repeating groups like “Tax” can occur up to the maximum number that is defined in the JSON schema. In the case of a repeating group, all occurrences shall be enclosed by square brackets [ ].

EXAMPLE 1 Repeating group Tax with only one occurrence:

```
"Tax": [{
        "TaxTypCd": "TAX01",
        "TaxLocAmt": 12.50,
        "GlTaxDbAcntNr": "ACC123456789",
        "GlTaxCrAcntNr": "ACC123456789"
    }],
```

EXAMPLE 2 Repeating group Tax with 4 occurrences:

```
"Tax": [{
        "TaxTypCd": "TAX01",
        "TaxLocAmt": 12.50,
        "GlTaxDbAcntNr": "ACC123456789",
        "GlTaxCrAcntNr": "ACC123456789"
    },
    {
        "TaxTypCd": "TAX02",
        "TaxLocAmt": 5.00,
        "GlTaxDbAcntNr": "ACC123456789",
        "GlTaxCrAcntNr": "ACC123456789"
    },
    {
        "TaxTypCd": "TAX03",
        "TaxLocAmt": 0.75,
        "GlTaxDbAcntNr": "ACC123456789",
        "GlTaxCrAcntNr": "ACC123456789"
    },
    {
        "TaxTypCd": "TAX04",
        "TaxLocAmt": 1.00,
        "GlTaxDbAcntNr": "ACC123456789",
        "GlTaxCrAcntNr": "ACC123456789"
    }],
```

4.3.2.6 Special characters to be escaped

The following characters are reserved characters and cannot be used in JSON and shall be properly replaced once they appear in strings:

backspace to be replaced with \b;

form feed to be replaced with \f;

newline to be replaced with \n;

carriage return to be replaced with \r;

tab to be replaced with \t;

double quote to be replaced with \";

backslash to be replaced with \\.

4.3.3 JSON schema

4.3.3.1 General

Each ADCS file is technically specified by a JSON schema and can also be validated by using that schema.

JSON schema files shall be created in accordance with the syntax specifications in ISO/IEC 21778, JSON Schema: A Media Type for Describing JSON Documents and also in accordance with the functional requirements written in ISO 21378.

Table 3 shows ISO 21378 representation formats and the equivalents used in JSON schema.

Table 3 — Representation specification in JSON schema

ISO 21378 representation	JSON schema representation
%ns	"type": "string", "maxLength": n,
%nc	"type": "string", "minLength": n, "maxLength": n,
%m.nf	"type": "number", "multipleOf": 0.0001, (when n = 4) "multipleOf": 0.00000001, (when n = 8)
%nd	"type": "number",

Table 4 shows ISO 21378 datatypes and the equivalents used in JSON schema.

Table 4 — Datatypes in JSON schema

ISO 21378 datatype	JSON schema datatype
Date	"type": "string", "minLength": 10, "maxLength": 10, "pattern": "^\\d{4}-(0[1-9]|1[0-2])-(0[1-9]|[1-2]\\d|3[0-1])$",
Time	"type": "string", "minLength": 8, "maxLength": 8, "pattern": "^([0-1]\\d|2[0-3]):([0-5]\\d):([0-5]\\d)$",
String	"type": "string",
Decimal	"type": "number",
Integer	"type": "number",
Boolean	"type": "boolean",

All schema files are bundled in one official ISO delivery package. Another development package is also available.

a) Official ISO delivery package: This package contains the schemas established in this document and ISO 21378 for exchanging the audit data.

b) Development package: This package contains the schemas that are used to develop new versions of the existing schemas and for the development of extensions.

The electronic version of these JSON schemas can be downloaded from the URL which can be found in Annex E.

4.3.3.2 Official ISO delivery package

All schema files in this package have one and the same version number, which can be found in the schema file itself.

The first delivery of the package has version number 1. After a certain period, if there are sufficient reasons for this, a new package of schemas will be delivered with a successor version number, for example, version number 2.

This package contains all ADCS table schemas. The name of the JSON schema file equals "Adc" followed by the ADCS table name and then followed by "_jsd6.json" like: "AdcApAdjustments_jsd6.json".

In this package, all schemas have their own datatype definitions.

A list of these JSON schemas can be found in Annex C.

EXAMPLE Part of "AdcApAdjustments_jsd6.json":

```
{
    "$schema": "http://json-schema.org/draft-06/schema#",
    "$id": "http://schemas.iso.org/AdcsML/Messages/AdcApAdjustments_jsd6-v1",
    "description": "JSON Definition of message hierarchy: ADC AP Adjustments v1.0",
    "modelRef":[
        "Message:              ADC AP Adjustments v1.0",
        "Generated at:         25-05-2022 16:53:42",
        "Organisation:         This schema is part of ISO 21377 Technical Specification"
    ],
    "type": "object",
    "additionalProperties": false,
    "properties":{
        "$id":{
        "type": "string",
        "format": "uri-reference"
        },
        "ApAdj":{
            "type": "array",
            "title": "AP ADJUSTMENT",
            "minItems": 0,
            "maxItems": 999999,
            "items":{
                  "type": "object",
                  "additionalProperties": false,
                  "properties":{

                     "AdjId":{
                     "$ref": "#/definitions/domains/TypId100",
                     "title": "Adjustment ID"
                  },

                 "AdjNr":{
                     "$ref": "#/definitions/domains/TypText100",
                     "title": "Adjustment Number"
                  }

                },
                "required": [ "AdjId" ]
            }
        }
    },
    "required": [ ],
    "definitions": {
        "domains": {

           "TypId100": {
               "type": "string",
               "maxLength": 100,
               "title": "IDENTIFIER 100"
           },

           "TypText100": {
               "type": "string",
               "maxLength": 100,
               "title": "TEXT 100"
           }
        }
```

```
    }
}
```

4.3.3.3 Development package

This package contains the schemas that are used to develop new versions of the existing schemas and for the development of extensions.

In this package each schema has its own version number, which can be found in the JSON schema file. In the first delivery of this package, all schemas have version 1.0 like “AdcApAdjustments_jsd6-v1.0.json”. When a schema changes, the minor version number is incremented each time (e.g. version number changes from 1.0 to 1.1).

After a certain period of time, if there are sufficient reasons to do this, a new official ISO delivery package will be delivered with the next successor major version number, for example, version number 2 (see 4.3.3.2); and then also a new development package will be delivered, in which all schemas will have version 2.0.

In this package, all schemas have their own datatype definitions.

EXAMPLE Part of “AdcApAdjustments_jsd6-v1.1.json”:

```
{
    "$schema": "http://json-schema.org/draft-06/schema#",
    "$id": "http://schemas.iso.org/AdcsML/Messages/AdcApAdjustments_jsd6-v1",
    "description": "JSON Definition of message hierarchy: ADC AP Adjustments v1.1",
    "modelRef": [
        "Message:             ADC AP Adjustments v1.1",
        "Generated at:        25-05-2022 16:53:42",
        "Organisation:        This schema is part of ISO 21377 Technical Specification"
    ],
    "type": "object",
    "additionalProperties": false,
    "properties": {
        "$id": {
        "type": "string",
        "format": "uri-reference"
        },
        "ApAdj": {
            "type": "array",
            "title": "AP ADJUSTMENT",
            "minItems": 0,
            "maxItems": 999999,
            "items": {
                "type": "object",
                "additionalProperties": false,
                "properties": {

                    "AdjId": {
                        "$ref": "#/definitions/domains/TypId100",
                        "title": "Adjustment ID"
                    },

                    "AdjNr": {
                        "$ref": "#/definitions/domains/TypText100",
                        "title": "Adjustment Number"
                    }

                },
                "required": [ "AdjId" ]
            }
        }
    },
    "required": [ ],
    "definitions": {
        "domains": {

            "TypId100": {
                "type": "string",
```

```
                "maxLength": 100,
                "title": "IDENTIFIER 100"
            },

            "TypText100": {
                "type": "string",
                "maxLength": 100,
                "title": "TEXT 100"
            }
        }
    }
}
```

4.3.4 JSON sample

The following example shows one JSON data file that contains only one record of an ADCS table:

```
{
    "$id": "http://schemas.iso.org/AdcsML/Messages/AdcApAdjustments_jsd6-v1",
    "ApAdj": [ {
        "AdjId": "AD1234",
        "AdjNr": "123456789",
        "AdjTypNm": "Finance charge.",
        "AdjDocNr": "123456789",
        "InvoiId": "I001",
        "JrnId": "JRN1",
        "FiscYr": "2020",
        "AcntingPer": "M1",
        "AdjDt": "2020-02-01",
        "SuplAcntId": "S00001",
        "GlDbAcntNr": "ACC123456789",
        "GlCrAcntNr": "ACC123456789",
        "AdjAmt": {
            "FuncAmt": 125.1255,
            "FuncCurCd": "USD",
            "TrAmt": 125.1255,
            "TrCurCd": "USD",
            "RprtAmt": 125.1255,
            "RprtCurCd": "USD",
            "LocAmt": 125.1255,
            "LocCurCd": "USD"
        },
        "Tax": [ {
            "TaxTypCd": "TAX01",
            "TaxLocAmt": 26.2763,
            "GlTaxDbAcntNr": "ACC123456789",
            "GlTaxCrAcntNr": "ACC123456789"
        },
{
            "TaxTypCd": "TAX02",
            "TaxLocAmt": 10.0000,
            "GlTaxDbAcntNr": "ACC123456789",
            "GlTaxCrAcntNr": "ACC123456789"
        } ],
        "Crea": {
            "UserId": "JD",
            "Dt": "2020-03-02",
            "Tm": "09:31:00"
        },
        "Aprv": {
            "UserId": "JD",
            "Dt": "2020-03-02",
            "Tm": "09:31:00"
        },
        "LstMdf": {
            "UserId": "JD",
            "Dt": "2020-03-02",
            "Tm": "09:31:00"
        },
        "BusSg": [ {
            "SqNr": "3",
```

```
            "BusSgCd": "BS001"
        } ]
    } ]
}
```

4.4 CSV

4.4.1 General

To keep the three exchange formats (XML, JSON, CSV) consistent, this subclause explains how to use the technical solutions in the CSV format. The specifications of these three technical exchange formats are defined in a consistent manner, so that conversion between formats is possible. These technical requirements for CSV are optional. If users do not need conversion to XML or JSON, it's allowed to just follow ISO 21378 base specification.

4.4.2 Technical requirements

4.4.2.1 General

A CSV file, with the filename ending in .csv, stores tabular data, including numbers and text, in plain text. In the CSV data file, each line of the file is a data record except the first line that contains data element tags, and each record consists of one or more fields, which are separated by commas, semicolons or other delimiters.

Only one table, with a number of lines, shall be sent in one CSV data file.

Such a file contains (schematic):

```
”data-element-tag”;”group-tag.data-element-tag”<line break>
”value”;”value”<line break>
”value”;”value”<optional line break>
```

Italicized text in the example shall be replaced by XML tags.

In the above example a semicolon is used as a column delimiter, however this can also be another character like comma (,) or pipe (|).

<line break> shall be replaced by the line break characters used on the target operating system (e.g. CR/LF on Windows and LF on Unix systems).

The last line can contain an optional line break, <optional line break>. Software processing ADCS data should skip empty lines in the ADCS CSV data files, in order to realize robust processing software.

If CR/LF occurs in the data, the field value that contains CR/LF shall be enclosed by double quotes as textual data.

4.4.2.2 Construction of column header tags

CSV data files use the XML tags as column header tags.

XML tags are in the first step derived from the field names, and in the second step shortened in a consistent manner, according to an abbreviation list (see Annex A).

4.4.2.3 Representation of (repeating) groups

Column tags of data elements which belong to a group should have the group tag followed by a dot “.” and their own tag. For instance, "FuncAmt" in group "Amt" should have column header tag "Amt. FuncAmt".

Column header tags of data elements which belong to a repeating group should be prefixed by the group tag plus “( )” that encloses a number ranging from 1 to the maximum repeat number, then be separated from the element tag by a dot “.”.

For instance, the repeating group “Tax”:

Tax(1).TaxTypeCd, Tax(1).TaxLocAmt, Tax(2).TaxTypeCd, Tax(2).TaxLocAmt, etc.

4.4.2.4 Numerical data

Numerical data can be exchanged without being enclosed by double quotes (“"). Numerical data can contain the dot character (.) as decimal separator. Other separators (for example thousand separator) are not allowed.

4.4.2.5 Special characters to be escaped

If Quotation mark (“) occurs in the data, it shall be doubled like “”.

4.4.2.6 Optional fields

Although certain fields or columns are not required in a CSV data file, the columns with their column headers should be always presented. The value of an optional field that is NULL can be presented as two adjacent column separators in the CSV data file (e.g. ;;).

4.4.3 CSV sample

In the following example of a CSV data file, there's one table with one header line and one record:

```
AdjId;AdjNr;AdjTypNm;AdjDocNr;InvoiId;JrnId;FiscYr;AcntingPer;AdjDt;SuplAcntId;GlDbA
cntNr;GlCrAcntNr;AdjAmt.FuncAmt;AdjAmt.FuncCurCd;AdjAmt.TrAmt;AdjAmt.TrCurCd;AdjAmt.
RprtAmt;AdjAmt.RprtCurCd;AdjAmt.LocAmt;AdjAmt.LocCurCd;Tax(1).TaxTypCd;Tax(1).
TaxLocAmt;Tax(1).GlTaxDbAcntNr;Tax(1).GlTaxCrAcntNr;Tax(2).TaxTypCd;Tax(2).
TaxLocAmt;Tax(2).GlTaxDbAcntNr;Tax(2).GlTaxCrAcntNr;Tax(3).TaxTypCd;Tax(3).
TaxLocAmt;Tax(3).GlTaxDbAcntNr;Tax(3).GlTaxCrAcntNr;Tax(4).TaxTypCd;Tax(4).
TaxLocAmt;Tax(4).GlTaxDbAcntNr;Tax(4).GlTaxCrAcntNr;Crea.UserId;Crea.Dt;Crea.Tm;Aprv.
UserId;Aprv.Dt;Aprv.Tm;LstMdf.UserId;LstMdf.Dt;LstMdf.Tm;BusSg(1).BusSgCd;BusSg(2).
BusSgCd;BusSg(3).BusSgCd

AD1234;123456789;Finance charge;123456789;I001;JRN1;2020;M1;2020-02-01;S00001;ACC12
3456789;ACC123456789;125.1255;USD;125.1255;USD;125.1255;USD;125.1255;USD;TAX01;12.5
0;ACC123456789;ACC123456789;TAX02;5.00;ACC123456789;ACC123456789;TAX03;0.75;ACC123456
789;ACC123456789;TAX04;1.00;ACC123456789;ACC123456789;JD;2020-03-02;09:31;JD;2020-03-
02;09:31;JD;2020-03-02;09:31;1000;1200;1234
```

Figure 1 shows the CSV example opened in OpenOffice Calc.

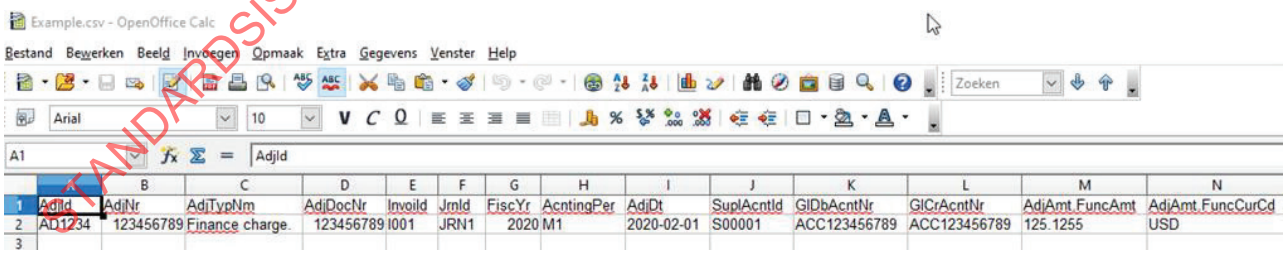


Figure 1 — CSV sample presented by OpenOffice Calc

4.5 Mapping ADCS tables to XML and JSON

4.5.1 General

This subclause specifies the mapping between full field names of ADCS tables and the tags of XML or JSON. One mapping table is set up for one ADCS table. So, 71 mapping tables are created.

4.5.2 Mapping table sample

Table 5 shows the way to map between full field names of the table AP_Adjustments and the tags of XML or JSON.

Table 5 — Mapping sample between ADCS table and XML or JSON

No.	Element name	Datatype	XML/JSON tag
1	Adjustment_ID	String	AdjId
2	Adjustment_Number	String	AdjNr
3	Adjustment_Type_Name	String	AdjTypNm
4	Adjustment_Document_Number	String	AdjDocNr
5	Invoice_ID	String	InvoiId
6	Journal_ID	String	JrnId
7	Fiscal_Year	String	FiscYr
8	Accounting_Period	String	AcctngPer
9	Adjustment_Date	Date	AdjDt
10	Supplier_Account_ID	String	SuplAcntId
11	Adjustment_Functional_Amount	Decimal	AdjAmt.FuncAmt
12	ADJ_Functional_CUR_Code	String	AdjAmt.FuncCurCd
13	ADJ_Transaction_Amount	Decimal	AdjAmt.TrAmt
14	ADJ_TRX_CUR_Code	String	AdjAmt.TrCurCd
15	Adjustment_Reporting_Amount	Decimal	AdjAmt.RprtAmt
16	Adjustment_Reporting_CUR_Code	String	AdjAmt.RprtCurCd
17	Adjustment_Local_Amount	Decimal	AdjAmt.LocAmt
18	Adjustment_Local_Currency_Code	String	AdjAmt.LocCurCd
19	Created_User_ID	String	Crea.UserId
20	Created_Date	Date	Crea.Dt
21	Created_Time	Time	Crea.Tm
22	Approved_User_ID	String	Aprv.UserId
23	Approved_Date	Date	Aprv.Dt
24	Approved_Time	Time	Aprv.Tm
25	Last_Modified_User_ID	String	LstMdf.UserId
26	Last_Modified_Date	Date	LstMdf.Dt
27	Last_Modified_Time	Time	LstMdf.Tm
28	Tax1_Type_Code	String	Tax(1).TaxTypCd
29	Tax1_Local_Amount	Decimal	Tax(1).TaxLocAmt
30	Tax2_Type_Code	String	Tax(2).TaxTypCd
31	Tax2_Local_Amount	Decimal	Tax(2).TaxLocAmt
32	Tax3_Type_Code	String	Tax(3).TaxTypCd
33	Tax3_Local_Amount	Decimal	Tax(3).TaxLocAmt
34	Tax4_Type_Code	String	Tax(4).TaxTypCd
35	Tax4_Local_Amount	Decimal	Tax(4).TaxLocAmt
36	GL_Debit_Account_Number	String	GlDbAcntNr
37	GL_Credit_Account_Number	String	GlCrAcntNr
38	GL_Tax1_Debit_Account_Number	String	Tax(1).GlTaxDbAcntNr
39	GL_Tax1_Credit_Account_Number	String	Tax(1).GlTaxCrAcntNr
40	GL_Tax2_Debit_Account_Number	String	Tax(2).GlTaxDbAcntNr

Table 5 *(continued)*

No.	Element name	Datatype	XML/JSON tag
41	GL_Tax2_Credit_Account_Number	String	Tax(2).GlTaxCrAcntNr
42	GL_Tax3_Debit_Account_Number	String	Tax(3).GlTaxDbAcntNr
43	GL_Tax3_Credit_Account_Number	String	Tax(3).GlTaxCrAcntNr
44	GL_Tax4_Debit_Account_Number	String	Tax(4).GlTaxDbAcntNr
45	GL_Tax4_Credit_Account_Number	String	Tax(4).GlTaxCrAcntNr
46	Business_Segment_X	String	BusSg(n).SqNr + BusSg(n).BusSgCd

4.6 Cross topics

4.6.1 General

This subclause specifies the common topics of these data formats, such as encryption of sensitive data.

4.6.2 Character encoding conventions

The character encoding used in ADCS should be developed independently from system and language (cross-platform and cross-language). Two parties exchanging files should adopt the UTF-8 format defined by the Unicode encoding system in the absence of a clear agreement. UTF-8 can encode all Unicode characters and is supported by many application systems. It has been also widely applied in many fields and is usually used as the default format for data exchange. Special agreements can be carried out in accordance with specific business practices. Examples:

ASCII can be used in English, Chinese national standard GB 18030-2005 can be used in Chinese.

ISO/IEC 8859-1 can be used in Western European languages.

4.6.3 Versioning

Versioning has two primary aspects.

a) Versioning of ADCS: Because the ADCS can evolve, it is important to identify the set of standards with which the extract file is compliant to avoid confusion when time flies and to increase the value of ADCS, particularly when an archived file set is leveraged for additional processing.

b) Versioning of data extracts: It is common for an organization to re-extract data for a specific task and period of time. An initial draft should be performed for testing purposes before the data is complete. This can help mitigate errors, omissions or other issues that require re-extracting the data. Therefore, it is important to easily differentiate between different versions of extracted files. In addition, it is also important to distinguish later extracts that replace or augment an original extract.

Over time, the standard can be refreshed to capture the types of changes referenced in the above aspects. For delimited text files, the file name is the primary tool for capturing these differences.

4.6.4 Multi-platform and multi-system data transfer

Different platforms, operating systems and file systems are likely to have an impact on file transfer.

Different file systems support different file sizes. For example, the maximum file size of the NTFS file system (in Windows) is 2 TB (terabytes), the maximum file size of the Fat32 file system (in Windows) is 4 GB (gigabytes), the maximum file size of the Ext2 file system (in Linux) is 16 GB, and the maximum file size of the Ext4 file system (in Linux) is 16 TB. If the auditor's and the auditee's file structures are different, direct data transfer can produce errors.

To ensure the efficiency and accuracy of data transfer, the following recommendations apply.

a) Considering the minimum system requirement, the limit of a single file size is 4 GB. Single files greater than 4 GB should be split into smaller files. The sequence number of the file is expressed in the file name (extracted table piece count), see 4.6.5.

b) When the transfer mode is binary conversion, files shall be transferred without modification.

EXAMPLE Changing the encoding to convert "\n" to "\r\n" while transferring files can corrupt the file and render it unreadable.

4.6.5 Conventions of folder naming and file naming

Because of the large number of files that can be involved in an extraction, naming conventions for folders and files are important. The conventions make files more easily distinguished from one another and easier to browse and identify. In the case of the ADCS, the following usage scenarios should be considered:

a) the organization of an auditee's files includes versions of files in a single data collection process or multiple data collections processes;

b) the differentiation of files across multiple auditees.

To satisfy the above scenarios, file information should be built into the directory structure and filenames.

The folder-naming conventions and some examples are shown in Table 6.

Table 6 — Folder-naming conventions and examples

Type of information	Information examples	Folder-naming examples
Company	<Widget, Inc>, <BU123>	<Widget, Inc_Retail_CHN_2018_ADCS_GL_1to9>
Division	<Retail>, <Manufacturing>, <Purchase>	
Country code	<USA>, <CHN>, <NL>, <23>	
Year	<2018>	
Standard	<ADCS>	
Module	<GL>, <INV>	
Periods	<Jan-Aug>, <1to9>	

The file-naming conventions and some examples are showed in Table 7.

Table 7 — File-naming conventions and examples

Type of information	Information examples	File-naming examples
Standard name	<ADCS>	<ADCS_7_2018_Widget, Inc_INV_INV_Location_20181108_1of1_00023.xml>
Period end extracted	<October>, <7>	
Year extracted	<2018>, <19>	
Company	<Widget, Inc>, <BU123>	
Module	<GL>, <INV>	
Table name	<SAL_Orders>, <INV_Location>	
Extraction date	<20181108>	
Extracted table piece count	<1of1>, <1of7>, <2of7>, <3of7>	
Extraction unique ID	<EID00001>, <XID1>, <00023>	
Extension	<xml>, <json>, <csv>, <txt>	

4.6.6 File compression

File compression is the representation of a source file with fewer bits, which can improve the data transfer efficiency and reduce storage consumption.

With ADCS data collection and exchange, in some scenarios, file compression is recommended as a way to exchange data. File compression conventions are as follows.

a) All the files and folders collected from the auditee are compressed into one single file.

b) The compression format should follow local convention.

EXAMPLE The ZIP compressed file format.

c) Encryption is recommended. Popular file compression software typically supports file encryption. As a result, the parties involved in data exchange may select one encryption method in accordance with the agreed method.

File-naming conventions for file compression and some examples are showed in Table 8.

Table 8 — Compression file-naming conventions and examples

Type of information	Information examples	File-naming examples
Company	<Widget, Inc>, <BU123>	<Widget, Inc_Retail_CHN_2018_ ADCS_1to9>
Division	<Retail>, <Manufacturing>, <Purchase>	
Country code	<USA>, <CHN>, <NL>, <23>	
Year	<2018>	
Standard	<ADCS>	
Periods	<Jan-Aug>, <1to9>	

4.6.7 Representation of special field types

The field types are defined in ADCS, in particular, the following general types shall apply.

Format of Date fields: YYYY-MM-DD (in accordance with ISO 8601-1). EXAMPLE: "2020-03-30".

Format of Time fields: hh:mm:ss (in accordance with ISO 8601-1). EXAMPLE: "20:30:00".

Format of Amounts, Prices, Percentages: have a maximum number of digits, and a number of decimals. These fields always need to be written with the required number of decimals defined and a decimal point (dot, "."). EXAMPLE "100.25". Leading zeros are not allowed. For example, "0123.50" is not allowed, which should be written as "123.50"; "00.05" is not allowed and should be written as "0.05".

Format of Boolean: Value can be 0, 1, false, true.

Percentage fields contain 1/100 percentage. For instance, 6 % should be written as "0.06" in the exchanged files.

4.6.8 Extension methodology and rules

In many cases, the current ADCS is expected to contain sufficiently data elements and that no specific sector data or locally required data will be added.

If data elements, tables or even modules are missing, they can be added to the next main version of ADCS.

If there is still a need to provide specific sector or locally desired data, it is possible to extend ADCS with new tables.

These extensions should be modelled as separate tables that are linked to the ADCS tables by using primary–foreign key relationships. These extensions shall be used to add data to the ADCS, not to replace already defined data elements.

An extension shall be defined in an extension specification. The specifications should be drawn up in accordance with the ADCS rules and naming conventions, in tabular form.

Extensions should be made publicly available to foster awareness and reuse.

Extensions, after proven to be successful, can be input to the ADCS maintenance process. In this process data elements defined in the extension can be moved to the original ADCS table in a new version of the standard.

4.6.9 Encryption of sensible data

Audit data is the company's business data rather than public data, which needs privacy protection, especially the sensible data involving personal information. Therefore, the privacy protection of sensitive data is an important part of ADCS. Data encryption is an important means to protect private data.

Data encryption is highly recommended. The parties involved in data exchange can select one encryption method in accordance with the agreed method.

Annex A
(informative)

XML tags abbreviation list

XML tags abbreviation list is given in Table A.1.

Table A.1 — XML tags abbreviation list

No.	Full name	Abbreviation
1	Abbreviation	Abrv
2	Academic	Acad
3	Account	Acnt
4	Accrual	Accrl
5	Accumulated	Accum
6	Acquisition	Acq
7	Addition	Add
8	Address	Adr
9	Adjusted	Adjd
10	Adjustment	Adj
11	After	Aft
12	Allocation	Alloc
13	Amount	Amt
14	Application	Apl
15	Approved	Aprv
16	Balance	Bal
17	Before	Bef
18	Beginning	Beg
19	Billing	Bill
20	Branch	Bra
21	Business	Bus
22	Cancellation	Cncl
23	Change	Chng
24	Characteristic	Char
25	Code	Cd
26	Contact	Cnt
27	Content	Cont
28	Contract	Contr
29	Corresponding	Corr
30	Costing	Cstng
31	Country	Cntry
32	Created	Crea
33	Credit	Cr
34	Currency	Cur
35	Customer	Cust

Table A.1 *(continued)*

No.	Full name	Abbreviation
36	Customized	Cstmzd
37	Date	Dt
38	Debit	Db
39	Default	Dft
40	Department	Dep
41	Depreciable	Dprcbl
42	Depreciation	Depre
43	Description	Dscr
44	Details	Dtls
45	Developer	Dvlpr
46	Discount	Dsont
47	Dispatch	Disp
48	Document	Doc
49	Employee	Emp
50	Employment	Emplmnt
51	Encoding	Enc
52	Ending	End
53	Exclude	Excl
54	Expense	Expns
55	External	Ext
56	Fiscal	Fisc
57	Functional	Func
58	Generated	Gen
59	Grouping	Grp
60	Header	Hdr
61	Hierarchy	Hrchy
62	Identifier	Id
63	Impairment	Impr
64	Include	Incl
65	Indicator	Ind
66	Inventory	Inv
67	Invoice	Invoi
68	Journal	Jrn
69	Last	Lst
70	Line	Ln
71	Local	Loc
72	Location	Lct
73	Materials	Mat
74	Measurement	Mea
75	Method	Mthd
76	Modified	Mdf
77	Module	Mod
78	Name	Nm
79	Number	Nr

Table A.1 *(continued)*

No.	Full name	Abbreviation
80	Open	Opn
81	Order	Ord
82	Organization	Org
83	Parent	Par
84	Payable	Pbl
85	Payment	Pay
86	Percentage	Perc
87	Period	Per
88	Person	Prsn
89	Physical	Phys
90	Primary	Prim
91	Proceeds	Prcds
92	Process	Proc
93	Product	Prdct
94	Project	Proj
95	Proportion	Prop
96	Province	Prvnc
97	Provision	Prov
98	Purchaser	Prchsr
99	Purchase	Pur
100	Purchasing	Prchsng
101	Quantity	Qt
102	Realized	Rlzd
103	Receipt	Rcpt
104	Receivable	Rcvbl
105	Received	Rcvd
106	Records	Rec
107	Reference	Ref
108	Regulator	Rgltr
109	Remaining	Rmng
110	Remark	Rmrk
111	Removal	Rmv
112	Replacement	Rplc
113	Reporting	Rprt
114	Requisition	Rqstn
115	Residual	Resi
116	Responsibility	Resp
117	Reversal	Rev
118	Sales	Sal
119	Segment	Sg
120	Service	Srvc
121	Settlement	Setl
122	Sequence	Sq
123	Sheet	Sht

Table A.1 *(continued)*

No.	Full name	Abbreviation
124	Shipment	Shp
125	Shipping	Shpng
126	Software	Sftw
127	Source	Src
128	Standard	Std
129	Status	Stat
130	Stocking	Stck
131	Subledger	Sbldgr
132	Supplier	Supl
133	System	Syst
134	Terms	Trms
135	Time	Tm
136	Total	Tot
137	Transaction	Tr
138	Type	Typ
139	Unrealized	Unrlzd
140	Version	Vers
141	Year	Yr

Annex B
(informative)

List of XML schemas

Adc_DataTypes.xsd

AdcBasAccountingPeriod.xsd

AdcBasBankAccount.xsd

AdcBasBillType.xsd

AdcBasBusinessSegment.xsd

AdcBasBusinessSegmentHierarchy.xsd

AdcBasChartOfAccounts.xsd

AdcBasCurrency.xsd

AdcBasCustomer.xsd

AdcBasCustomerType.xsd

AdcBasCustomizedAccSegment.xsd

AdcBasCustomizedAccValue.xsd

AdcBasEmployee.xsd

AdcBasJournalEntryType.xsd

AdcBasMeasurementUnit.xsd

AdcBasPaymentTerm.xsd

AdcBasProfile.xsd

AdcBasProject.xsd

AdcBasSettlementMethod.xsd

AdcBasSupplier.xsd

AdcBasSupplierType.xsd

AdcBasTaxRegulatory.xsd

AdcBasTaxType.xsd

AdcBasUser.xsd

AdcGlAccountSegment.xsd

AdcGlAccountsPeriodBalance.xsd

AdcGlDetails.xsd

AdcGlSource.xsd

AdcGlTrialBalance.xsd

AdcArAdjustments.xsd

AdcArAdjustmentsDetails.xsd

AdcArCashApplication.xsd

AdcArCashReceived.xsd

AdcArOpenAccountsReceivable.xsd

AdcSalContracts.xsd

AdcSalContractsDetails.xsd

AdcSalInvoicesGenerated.xsd

AdcSalInvoicesGeneratedDetails.xsd

AdcSalOrders.xsd

AdcSalOrdersDetails.xsd

AdcSalShipmentsMade.xsd

AdcSalShipmentsMadeDetails.xsd

AdcApAdjustments.xsd

AdcApAdjustmentsDetails.xsd

AdcApCashApplication.xsd

AdcApOpenAccountsPayable.xsd

AdcApPaymentsMade.xsd

AdcPurContracts.xsd

AdcPurContractsDetails.xsd

AdcPurInvoicesReceived.xsd

AdcPurInvoicesReceivedDetails.xsd

AdcPurMaterialsReceived.xsd

AdcPurMaterialsReceivedDetails.xsd

AdcPurOrders.xsd

AdcPurOrdersDetails.xsd

AdcPurRequisitions.xsd

AdcPurRequisitionsDetails.xsd

AdcInvLocation.xsd

AdcInvOnHand.xsd

AdcInvPeriodBalance.xsd

AdcInvPhysicalInventory.xsd

AdcInvProduct.xsd

AdcInvProductType.xsd

AdcInvTransaction.xsd

AdcPpeAddition.xsd

AdcPpeChange.xsd

AdcPpeDepartmentAllocation.xsd

AdcPpeDepreciation.xsd

AdcPpeDepreciationMethod.xsd

AdcPpeMaster.xsd

AdcPpeRemoval.xsd

AdcPpeType.xsd

Annex C
(informative)

List of JSON schemas

AdcBasAccountingPeriod_jsd6.json

AdcBasBankAccount_jsd6.json

AdcBasBillType_jsd6.json

AdcBasBusinessSegment_jsd6.json

AdcBasBusinessSegmentHierarchy_jsd6.json

AdcBasChartOfAccounts_jsd6.json

AdcBasCurrency_jsd6.json

AdcBasCustomer_jsd6.json

AdcBasCustomerType_jsd6.json

AdcBasCustomizedAccSegment_jsd6.json

AdcBasCustomizedAccValue_jsd6.json

AdcBasEmployee_jsd6.json

AdcBasJournalEntryType_jsd6.json

AdcBasMeasurementUnit_jsd6.json

AdcBasPaymentTerm_jsd6.json

AdcBasProfile_jsd6.json

AdcBasProject_jsd6.json

AdcBasSettlementMethod_jsd6.json

AdcBasSupplier_jsd6.json

AdcBasSupplierType_jsd6.json

AdcBasTaxRegulatory_jsd6.json

AdcBasTaxType_jsd6.json

AdcBasUser_jsd6.json

AdcGlAccountSegment_jsd6.json

AdcGlAccountsPeriodBalance_jsd6.json

AdcGlDetails_jsd6.json

AdcGlSource_jsd6.json

AdcGlTrialBalance_jsd6.json

AdcArAdjustments_jsd6.json

AdcArAdjustmentsDetails_jsd6.json

AdcArCashApplication_jsd6.json

AdcArCashReceived_jsd6.json

AdcArOpenAccountsReceivable_jsd6.json

AdcSalContracts_jsd6.json

AdcSalContractsDetails_jsd6.json

AdcSalInvoicesGenerated_jsd6.json

AdcSalInvoicesGeneratedDetails_jsd6.json

AdcSalOrders_jsd6.json

AdcSalOrdersDetails_jsd6.json

AdcSalShipmentsMade_jsd6.json

AdcSalShipmentsMadeDetails_jsd6.json

AdcApAdjustments_jsd6.json

AdcApAdjustmentsDetails_jsd6.json

AdcApCashApplication_jsd6.json

AdcApOpenAccountsPayable_jsd6.json

AdcApPaymentsMade_jsd6.json

AdcPurContracts_jsd6.json

AdcPurContractsDetails_jsd6.json

AdcPurInvoicesReceived_jsd6.json

AdcPurInvoicesReceivedDetails_jsd6.json

AdcPurMaterialsReceived_jsd6.json

AdcPurMaterialsReceivedDetails_jsd6.json

AdcPurOrders_jsd6.json

AdcPurOrdersDetails_jsd6.json

AdcPurRequisitions_jsd6.json

AdcPurRequisitionsDetails_jsd6.json

AdcInvLocation_jsd6.json

AdcInvOnHand_jsd6.json

AdcInvPeriodBalance_jsd6.json

AdcInvPhysicalInventory_jsd6.json

AdcInvProduct_jsd6.json

AdcInvProductType_jsd6.json

AdcInvTransaction_jsd6.json

AdcPpeAddition_jsd6.json

AdcPpeChange_jsd6.json

AdcPpeDepartmentAllocation_jsd6.json

AdcPpeDepreciation_jsd6.json

AdcPpeDepreciationMethod_jsd6.json

AdcPpeMaster_jsd6.json

AdcPpeRemoval_jsd6.json

AdcPpeType_jsd6.json

Annex D
(informative)

Data model

D.1 General

To design the XML schemas and JSON schemas in a consistent and structured way, a data model is created from ISO 21378 according to ISO 11179-1.

This data model consists of 3 parts:

a) object model;

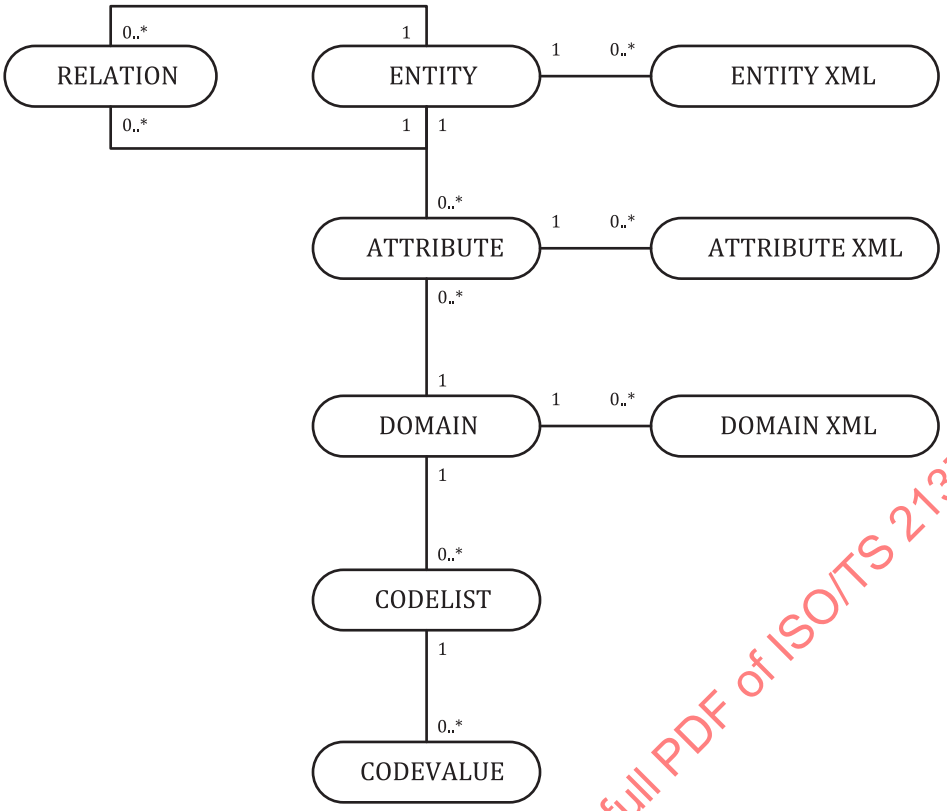
b) transaction model based on the object model;

c) message models based on the transaction model and the basis for exchange formats.

D.2 Object model

D.2.1 Structure

The object model consists of a collection of entities, attributes, domains, code-lists, code-values and XML specific properties. This object model is structured according to the meta model shown in Figure D.1.



Key

Components

0…* cardinality zero to many

1 cardinality one

Connections and lines

relationship between objects

Figure D.1 — Meta data model diagram for the object model

D.2.2 Description of the different components

D.2.2.1 Entity

An entity has a name and a description that records data. In this way, an entity describes an object.

Examples of entities are “SAL Order”, “BAS Customer”, “INV Product”.

An entity can be a further division from another entity. An example is the division of 'Address' into 'Physical Address' and 'Billing Address' (specializations). The "Physical Address" entity is the so-called subtype of the "Address" entity. This latter entity is called the super-type. Super and subtypes have their own attributes, and the subtypes also inherit the attributes of the super-type. The subtype has a so-called “is_a” relationship with the super-type.

D.2.2.2 Relation

A relation is the relationship property between two entities.

D.2.2.3 Attribute

An attribute describes one property of a single entity. Examples of attributes are “Fiscal Year”, “GL Account Number”, “Tax Percentage”.