



ISO/TC 295/WG 1 "Audit Data Collection for Non-financial Enterprises"

Convenorship: SAC

Convenor: WANG Wenyu Ms



20210610_Exchange formats for Audit Data Collection Standard XML and JSON

Document type	Related content	Document date	Expected action
Meeting / Working documents for discussion	Project: ISO/WD TS 5409	2021-06-15	INFO

Exchange formats for the Audit Data Collection Standard: XML and JSON

WD stage

Warning for WDs and CDs

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

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.

To help you, this guide on writing standards was produced by the ISO/TMB and is available at <https://www.iso.org/iso/how-to-write-standards.pdf>

A model manuscript of a draft International Standard (known as “The Rice Model”) is available at https://www.iso.org/iso/model_document-rice_model.pdf

© ISO 2021

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO Copyright Office

CP 401 • CH-1214 Vernier, Geneva

Phone: + 41 22 749 01 11

Fax: + 41 22 749 09 47

Email: copyright@iso.org

Website: www.iso.org

Published in Switzerland

Contents

Foreword	iv
Introduction	v
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Exchange Formats	4
4.1 General	4
4.2 XML	5
4.2.1 General	5
4.2.2 Technical guideline	5
4.2.3 XML Schemas	6
4.2.4 XML Sample	19
4.3 JSON	21
4.3.1 General	21
4.3.2 Technical guideline	21
4.3.3 JSON Schema	23
4.3.4 JSON Sample message	28
4.4 CSV	31
4.4.1 General	31
4.4.2 Technical guideline	31
4.4.3 CSV validation	32
4.4.4 CSV Sample	32
4.5 Mapping ISO 21378:2019 ADCS Tables to XML and JSON	33
4.5.1 Mapping Table sample	33
4.6 Cross topics	35
4.6.1 General	35
4.6.2 Character encoding conventions	35
4.6.3 Versioning	35
4.6.4 Multi-platform and multi-system data transfer	35
4.6.5 Limiting file size	36
4.6.6 File Compression	36
4.6.7 Conventions of folder naming and file naming	36
4.6.8 Representation of special field types	37
4.6.9 Extension methodology and rules	37
4.6.10 Encryption of sensible data	38
Annex A XML Tags abbreviation List	39
Annex B List of XML Schemas	43
Annex C List of JSON Schemas	45
Annex D Data Model	47
Bibliography	54

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 Project Committee ISO/PC 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 the international standard Audit Data Collection ISO 21378:2019.

Exchange formats for Audit Data Collection Standard:

XML and JSON

1 Scope

The Audit Data Collection ISO 21378:2019 (ADCS) defines the functional requirements for exchanging audit data in flat file format. This document concerns the specification of technical exchange formats in XML, JSON and flat file (CSV) as output file formats for the functional content defined in the international standard Audit Data Collection ISO 21378:2019.

Besides this document the deliverable contains per ADCS Table:

- a. XML Schema
- b. XML Sample file
- c. JSON Schema
- d. JSON Sample file

To keep the three exchange formats (XML, JSON, CSV) consistent, we will also explain 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/IEC 11179-1:2015, *Information technology — Metadata registries (MDR)*

REC-xml-19980210, *Extensible Markup Language (XML) 1.0 W3C Recommendation 10-February-1998*

XML Schema 1.1 Part 1: *Structures*

XML Schema 1.1 Part 2: *Datatypes*

RFC4627, *The application/json Media Type for JavaScript Object Notation (JSON)*

JSON Schema definition specifications on: <https://json-schema.org/specification-links.html>

3 Terms and definitions

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

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

- a. ISO Online browsing platform: available at <https://www.iso.org/obp>
- b. IEC Electropedia: available at <http://www.electropedia.org/>

3.1

data

set of values of qualitative or quantitative variables

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 14825:2011, 3.1.2, modified — The word "related" and the note to entry have been deleted.]

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 — "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

A data model contains formal and normalized descriptions of the data requirements. It consists of entities, relations, attributes and domains.

3.7

entity

is a group of data elements describing an object. Equivalent to "Object Class" in ISO 11179.

3.8

relation

relationship between two *entities* (3.7)

3.9

attribute

data element (3.2) describing an object. Equivalent to "Property" in ISO 11179.

3.10

domain

set of properties to define the value space of attributes. Can contain code lists and code values. Equivalent to "Representation" in ISO 11179.

3.11

code list

standardised 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 case of ADCS, three exchange formats are specified:

1. XML files defined by XML Schema specification (W3C),
2. JSON files defined JSON Schema (Version 0.6) specification,
3. CSV files.

These three formats will contain the same functional ADCS content specification but will differ on the technical level only.

This means that these exchange formats are convertible from one to the other. For instance, you could convert XML files to CSV files or CSV files to JSON files.

To generate the XML and JSON schemas in a consistent way, a data model was set up from ISO 21378:2019 Audit data collection according to ISO/IEC 11179-1:2015. 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”, e.g. “Physical Address”, “Billing Address”, “Tax”, “Created”, “Modified”, “Posted” etc., completely in line with the ISO 21378:2019 Audit data collection standard. This way of working guarantees that all reusable groups that occur in multiple tables will be defined the same way.

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

In the following chapters the various technical specifications are explained, and the additional packaging and communication agreements are defined.

4.2 XML

4.2.1 General

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

4.2.2 Technical guideline

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

The filename must comply with the rules described in chapter 4.6.7.

Each ADCS table has its own XML-Schema to define its XML-file. So, when there are 71 tables, there are 71 XML-Schemas.

Such an XML 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>
```

Italics printed text should be replaced by the definitions in this document.

4.2.2.1 Root tag

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

Used table names are taken from the ISO 21378:2019 standard.

4.2.2.2 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"

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

Because ADCS are bulk files by nature, abbreviated tags will reduce the file sizes by 30% - 40%.

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

In XML Schemas the full data element names are documented as annotation.

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

4.2.2.4 Optional fields

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

4.2.2.5 Repeating groups

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

4.2.2.6 Special characters to be escaped

<	<	less than
>	>	greater than
&	&	ampersand
'	'	apostrophe
"	"	quotation mark

4.2.3 XML Schemas

Each ADCS file is technically specified by an XML Schema and can also be validated by using that schema. All schema files are bundled in a schema package.

The following schema packages are available:

1. Official ISO delivery package: This package contains the schemas that have been officially established as the ISO standard, and should be used to communicate the audit data in a standardized manner.
2. Development package: This package contains the schemas that are used to develop new versions of the existing schemas and for the development of extension schemas.

The following chapters explain the two types of packages.

4.2.3.1 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 will have 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:

1. All ADCS table schemas where the name of the schema file is equal to the root tag name like “AdcApAdjustments.xsd”.
2. One data types schema, named “ADC_DataTypes.xsd”. This schema is imported into all ADCS table schemas.

Example: “AdcApAdjustments.xsd”:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- XML Definition of message hierarchy -->
<!-- Message: ADC AP Adjustments v1.0 -->
<!-- Message Rev.: 10 -->
<!-- Transaction: AA - Audit Data Collection -->
<!-- Transaction Rev.: 271 -->
<!-- Datamodel: Audit Data Collection Datamodel, version HEAD -->
<!-- Datamodel Rev.: 347 -->
<!-- Definition type: W3C Schema (http://www.w3.org/XML/Schema) -->
<!-- XML tagset used: AdcsML -->
<!-- Generated at: 02-05-2021 16:00:55 -->
<!-- NOTE: This schema was generated automatically: manual modifications will be lost -->
<!-- NOTE: Additional rules/conditions regarding the definition may exist, see functional hierarchical 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:element name="AdjTypNm" type="cdt:TypText60" minOccurs="0" maxOccurs="1">
                <xsd:annotation>
                  <xsd:documentation>Adjustment Type Name</xsd:documentation>
                </xsd:annotation>
              </xsd:element>
              <xsd:element name="AdjDocNr" type="cdt:TypText100" minOccurs="0" maxOccurs="1">
                <xsd:annotation>
```

```

        <xsd:documentation>Adjustment Document Number</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="InvoId" type="cdt:TypId60" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Invoice ID</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="JrnId" type="cdt:TypId100" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Journal ID</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="FiscYr" type="cdt:TypYrId" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Fiscal Year</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="AcntingPer" type="cdt:TypPerId" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Accounting Period</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="AdjDt" type="cdt:TypDt" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Adjustment Date</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="SuplAcntId" type="cdt:TypId100" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Supplier Account ID</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="GIDbAcntNr" type="cdt:TypGIAcntNr100" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>GL Debit Account Number</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="GICrAcntNr" type="cdt:TypGIAcntNr100" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>GL Credit Account Number</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="TrAmt" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>ap adjustment - TRANSACTION AMOUNT</xsd:documentation>
      </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="FuncAmt" type="cdt:TypAmt" minOccurs="0" maxOccurs="1">
          <xsd:annotation>
            <xsd:documentation>Functional Amount</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="FuncCurCd" type="cdt:TypCurCd" minOccurs="0" maxOccurs="1">
          <xsd:annotation>
            <xsd:documentation>Functional Currency Code</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:sequence>
</xsd:element>

```

```

        </xsd:annotation>
      </xsd:element>
      <xsd:element name="TrAmt" type="cdt:TypAmt" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Transaction Amount</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="TrCurCd" type="cdt:TypCurCd" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Transaction Currency Code</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="RprtAmt" type="cdt:TypAmt" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Reporting Amount</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="RprtCurCd" type="cdt:TypCurCd" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Reporting Currency Code</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="LocAmt" type="cdt:TypAmt" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Local Amount</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="LocCurCd" type="cdt:TypCurCd" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Local Currency Code</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Tax" minOccurs="0" maxOccurs="4">
  <xsd:annotation>
    <xsd:documentation>ap adjustment - TAX</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="TaxTypCd" type="cdt:TypTaxTypCd" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Tax Type Code</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="TaxLocAmt" type="cdt:TypAmt" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Tax Local Amount</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="GITaxDbAcntNr" type="cdt:TypGIAcntNr100" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>GL Tax Debit Account Number</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="GITaxCrAcntNr" type="cdt:TypGIAcntNr100" minOccurs="0" maxOccurs="1">

```



```

        <xsd:annotation>
          <xsd:documentation>GL Tax Credit Account Number</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Crea" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>ap adjustment - CREATED</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="UserId" type="cdt:TypId25" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>User ID</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="Dt" type="cdt:TypDt" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Date</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="Tm" type="cdt:TypTm" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Time</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Aprv" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>ap adjustment - APPROVED</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="UserId" type="cdt:TypId25" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>User ID</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="Dt" type="cdt:TypDt" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Date</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="Tm" type="cdt:TypTm" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Time</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="LstMdf" minOccurs="0" maxOccurs="1">
  <xsd:annotation>

```

```
<xsd:documentation>ap adjustment - LAST MODIFIED</xsd:documentation>
</xsd:annotation>
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="UserId" type="cdt:TypId25" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>User ID</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="Dt" type="cdt:TypDt" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Date</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="Tm" type="cdt:TypTm" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Time</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="BusSg" minOccurs="0" maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation>ap adjustment - BUSINESS SEGMENT</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="BusSgCd" type="cdt:TypId25" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Business Segment Code</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- XML Definition of elementary types -->
<!-- Datamodel:      Audit Data Collection Datamodel, version HEAD -->
<!-- Datamodel Rev.: 317 -->
<!-- Definition type: W3C Schema (http://www.w3.org/XML/Schema) -->
<!-- XML tagset used: AdcsML -->
<!-- Namespace prefix: cdt -->
<!-- Generated at:   15-04-2021 09:52:36 -->
<!-- NOTE:   This schema was generated automatically: manual modifications will be lost -->
<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">
<!-- NAME 25: FORMAT: AN..25 -->
<xsd:simpleType name="TypNm25">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="25"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- BUSINESS SEGMENT REFERENCE LEVEL: FORMAT: AN..2 -->
<xsd:simpleType name="TypBusSgRefLevel">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="2"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- NAME 60: FORMAT: AN..60 -->
<xsd:simpleType name="TypNm60">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="60"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- NAME 100: FORMAT: AN..100 -->
<xsd:simpleType name="TypNm100">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="100"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- TEXT 60: FORMAT: AN..60 -->
<xsd:simpleType name="TypText60">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="60"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- DATE: FORMAT: AN10 -->
<xsd:simpleType name="TypDt">
  <xsd:restriction base="xsd:date"/>
</xsd:simpleType>
<!-- IDENTIFIER 25: FORMAT: AN..25 -->
<xsd:simpleType name="TypId25">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="25"/>
  </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>
<!-- IDENTIFIER 60: FORMAT: AN..60 -->
<xsd:simpleType name="TypId60">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="60"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- CUSTOMER TYPE CODE: FORMAT: AN..100 -->
<xsd:simpleType name="TypCustTypCd">

```

```

    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!-- NAME 80: FORMAT: AN..80 -->
  <xsd:simpleType name="TypNm80">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="80"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!-- IDENTIFIER 100: FORMAT: AN..100 -->
  <xsd:simpleType name="TypId100">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!-- TAX IDENTIFICATION NUMBER: FORMAT: AN..100 -->
  <xsd:simpleType name="TypTaxIdentificationNr">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!-- CITY NAME: FORMAT: AN..100 -->
  <xsd:simpleType name="TypCityNm">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!-- STATE PROVINCE CODE: FORMAT: AN..6 -->
  <xsd:simpleType name="TypStatePrvncCd">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="6"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!-- POSTAL CODE: FORMAT: AN..20 -->
  <xsd:simpleType name="TypPostalCd">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="20"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!-- COUNTRY CODE: FORMAT: AN..3 -->
  <xsd:simpleType name="TypCntryCd">
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="2"/>
      <xsd:maxLength value="3"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!-- PHONE NUMBER: FORMAT: AN..20 -->
  <xsd:simpleType name="TypPhoneNr">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="20"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!-- EMAIL: FORMAT: AN..100 -->
  <xsd:simpleType name="TypEmail">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="100"/>
    </xsd:restriction>
  </xsd:simpleType>

```

```

    </xsd:restriction>
</xsd:simpleType>
<!-- AMOUNT: FORMAT: N..22,4 -->
<xsd:simpleType name="TypAmt">
    <xsd:restriction base="xsd:decimal">
        <xsd:totalDigits value="22"/>
        <xsd:fractionDigits value="4"/>
    </xsd:restriction>
</xsd:simpleType>
<!-- PERCENTAGE: FORMAT: N..5,4 -->
<xsd:simpleType name="TypPerc">
    <xsd:restriction base="xsd:decimal">
        <xsd:totalDigits value="5"/>
        <xsd:fractionDigits value="4"/>
    </xsd:restriction>
</xsd:simpleType> .....

```

4.2.3.2 Development package

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

In this package each schema has its own version number, which is used 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 will have version 1.0 like "AdcApAdjustments-v1.0.xsd". When a schema changes, the minor version number is incremented each time such a schema is distributed (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 of each other.

After a certain period of time, if there are sufficient reasons to do this, a new ISO package of schemas will be delivered with the next successor major version number, for example version number 2 (see chapter 4.2.3.1), and then also a new development package will be delivered, in which all schemas will have version 2.0.

Example: "AdcApAdjustments-v1.0.xsd":

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- XML Definition of message hierarchy -->
<!-- Message: ADC AP Adjustments Details v1.0 -->
<!-- Message Rev.: 5 -->
<!-- Transaction: AA - Audit Data Collection -->
<!-- Transaction Rev.: 271 -->
<!-- Datamodel: Audit Data Collection Datamodel, version HEAD -->
<!-- Datamodel Rev.: 347 -->
<!-- Definition type: W3C Schema (http://www.w3.org/XML/Schema) -->
<!-- XML tagset used: AdcsML -->
<!-- Generated at: 02-05-2021 16:00:56 -->
<!-- NOTE: This schema was generated automatically: manual modifications will be lost -->
<!-- NOTE: Additional rules/conditions regarding the definition may exist, see functional hierarchical specification -->
<xsd:schema xmlns:str="http://schemas.iso.org/AdcsML/Messages/AdcApAdjustmentsDetails-v1"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://schemas.iso.org/AdcsML/Messages/AdcApAdjustmentsDetails-v1"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="1.0">
  <xsd:element name="AdcApAdjustmentsDetails">
    <xsd:annotation>

```

```

    <xsd:documentation>MESSAGE</xsd:documentation>
  </xsd:annotation>
</xsd:complexType>
<xsd:sequence>
  <xsd:element name="ApAdjDtls" minOccurs="0" maxOccurs="unbounded">
    <xsd:annotation>
      <xsd:documentation>AP ADJUSTMENT DETAILS</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="AdjLnId" type="str:TypId60" minOccurs="1" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Adjustment Line ID</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="AdjLnNr" type="str:TypText10" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Adjustment Line Number</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="InvoId" type="str:TypId60" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Invoice ID</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="InvoiLnId" type="str:TypId60" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Invoice Line ID</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="JrnId" type="str:TypId100" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>Journal ID</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="GIDbAcntNr" type="str:TypGIAcntNr100" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>GL Debit Account Number</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="GICrAcntNr" type="str:TypGIAcntNr100" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>GL Credit Account Number</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="TrAmt" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>ap adjustment details - TRANSACTION AMOUNT</xsd:documentation>
      </xsd:annotation>
    </xsd:complexType>
    <xsd:sequence>
      <xsd:element name="FuncAmt" type="str:TypAmt" minOccurs="0" maxOccurs="1">

```

```

        <xsd:annotation>
          <xsd:documentation>Functional Amount</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="FuncCurCd" type="str:TypCurCd" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Functional Currency Code</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="TrAmt" type="str:TypAmt" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Transaction Amount</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="TrCurCd" type="str:TypCurCd" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Transaction Currency Code</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="RprtAmt" type="str:TypAmt" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Reporting Amount</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="RprtCurCd" type="str:TypCurCd" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Reporting Currency Code</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="LocAmt" type="str:TypAmt" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Local Amount</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="LocCurCd" type="str:TypCurCd" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Local Currency Code</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Tax" minOccurs="0" maxOccurs="4">
  <xsd:annotation>
    <xsd:documentation>ap adjustment details - TAX</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="TaxTypCd" type="str:TypTaxTypCd" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Tax Type Code</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="TaxLocAmt" type="str:TypAmt" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Tax Local Amount</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

```

```

</xsd:element>
<xsd:element name="GITaxDbAcntNr" type="str:TypGIAcntNr100" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>GL Tax Debit Account Number</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="GITaxCrAcntNr" type="str:TypGIAcntNr100" minOccurs="0" maxOccurs="1">
  <xsd:annotation>
    <xsd:documentation>GL Tax Credit Account Number</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="BusSg" minOccurs="0" maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation>ap adjustment details - BUSINESS SEGMENT</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="BusSgCd" type="str:TypId25" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Business Segment Code</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<!-- AMOUNT: FORMAT: N..22,4 -->
<xsd:simpleType name="TypAmt">
  <xsd:restriction base="xsd:decimal">
    <xsd:totalDigits value="22"/>
    <xsd:fractionDigits value="4"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- CURRENCY CODE: FORMAT: AN3 -->
<xsd:simpleType name="TypCurCd">
  <xsd:annotation>
    <xsd:documentation>ISO 4217, codes for representation of currencies.</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:length value="3"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- GL ACCOUNT NUMBER 100: FORMAT: AN..100 -->
<xsd:simpleType name="TypGIAcntNr100">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="100"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- IDENTIFIER 100: FORMAT: AN..100 -->

```



```

<xsd:simpleType name="TypId100">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="100"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- IDENTIFIER 25: FORMAT: AN..25 -->
<xsd:simpleType name="TypId25">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="25"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- IDENTIFIER 60: FORMAT: AN..60 -->
<xsd:simpleType name="TypId60">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="60"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- TAX TYPE CODE: FORMAT: AN..25 -->
<xsd:simpleType name="TypTaxTypCd">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="25"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- TEXT 10: FORMAT: AN..10 -->
<xsd:simpleType name="TypText10">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="10"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:schema>

```

4.2.4 XML Sample

In the following example there's one ADCS table with one line in one XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<AdcApAdjustments xsi:schemaLocation="http://schemas.iso.org/AdcsML/Messages/AdcApAdjustments-v1 AdcApAdjustments-v1.0.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>
    <Invoild>I001</Invoild>
    <JrnId>JRN1</JrnId>
    <FiscYr>2020</FiscYr>
    <AcntingPer>M1</AcntingPer>
    <AdjDt>2020-02-01</AdjDt>
    <SuplAcntId>S00001</SuplAcntId>
    <GIDbAcntNr>ACC123456789</GIDbAcntNr>
    <GICrAcntNr>ACC123456789</GICrAcntNr>
    <TrAmt>
      <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>
    </TrAmt>
    <Tax>
      <TaxTypCd>TAX01</TaxTypCd>
      <TaxLocAmt>12.50</TaxLocAmt>
      <GITaxDbAcntNr>ACC123456789</GITaxDbAcntNr>
      <GITaxCrAcntNr>ACC123456789</GITaxCrAcntNr>
    </Tax>
    <Tax>
      <TaxTypCd>TAX02</TaxTypCd>
      <TaxLocAmt>5.00</TaxLocAmt>
      <GITaxDbAcntNr>ACC123456789</GITaxDbAcntNr>
      <GITaxCrAcntNr>ACC123456789</GITaxCrAcntNr>
    </Tax>
    <Tax>
      <TaxTypCd>TAX03</TaxTypCd>
      <TaxLocAmt>0.75</TaxLocAmt>
      <GITaxDbAcntNr>ACC123456789</GITaxDbAcntNr>
      <GITaxCrAcntNr>ACC123456789</GITaxCrAcntNr>
    </Tax>
    <Tax>
      <TaxTypCd>TAX04</TaxTypCd>
      <TaxLocAmt>1.00</TaxLocAmt>
      <GITaxDbAcntNr>ACC123456789</GITaxDbAcntNr>
      <GITaxCrAcntNr>ACC123456789</GITaxCrAcntNr>
    </Tax>
    <Crea>
      <UserId>JD</UserId>
    </Crea>
  </ApAdj>
</AdcApAdjustments>
```

```

    <Dt>2020-03-02</Dt>
    <Tm>09:31</Tm>
  </Crea>
  <Aprv>
    <UserId>JD</UserId>
    <Dt>2020-03-02</Dt>
    <Tm>09:31</Tm>
  </Aprv>
  <LstMdf>
    <UserId>JD</UserId>
    <Dt>2020-03-02</Dt>
    <Tm>09:31</Tm>
  </LstMdf>
  <BusSg>
    <BusSgCd>1000</BusSgCd>
  </BusSg>
  <BusSg>
    <BusSgCd>1200</BusSgCd>
  </BusSg>
  <BusSg>
    <BusSgCd>1234</BusSgCd>
  </BusSg>
</ApAdj>
</AdcApAdjustments>

```

4.3 JSON

4.3.1 General

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

4.3.2 Technical guideline

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

The filename must comply with the rules described in chapter 4.6.7.

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 file contains (schematic):

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

Italics printed text should be replaced by the definitions in this document and JSON tag definitions from ADCS data catalog.

4.3.2.1 Root tag

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

Used table names are taken from the ISO 21378:2019 standard.

Remark: At this stage of the project it was not yet decided to use a root tag in JSON.

4.3.2.2 Target name space definition

When required in JSON, the XML target name space will be used.

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"

Remark: At this stage of the project it was not yet decided to use a target name space in JSON.

4.3.2.3 JSON tags

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

JSON uses the XML tags.

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

Because ADCS are bulk files by nature, abbreviated tags will reduce the file sizes by 30% - 40%.

Full data element names and abbreviated XML tags are together specified in ADCS to XML mapping tables (see for explanation and example of mapping tables chapter 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 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 case of a repeating group all occurrences in total will be enclosed by square brackets [].

Example 1: Repeating group Tax with only one occurrence:

```
"Tax": [{
  "TaxTypCd": "TAX01",
  "TaxLocAmt": 12.50,
  "GITaxDbAcntNr": "ACC123456789",
  "GITaxCrAcntNr": "ACC123456789"
}],
```

Example 2: Repeating group Tax with 4 occurrences:

```
"Tax": [{
  "TaxTypCd": "TAX01",
  "TaxLocAmt": 12.50,
  "GITaxDbAcntNr": "ACC123456789",
  "GITaxCrAcntNr": "ACC123456789"
},
{
  "TaxTypCd": "TAX02",
  "TaxLocAmt": 5.00,
  "GITaxDbAcntNr": "ACC123456789",
  "GITaxCrAcntNr": "ACC123456789"
},
{
  "TaxTypCd": "TAX03",
  "TaxLocAmt": 0.75,
  "GITaxDbAcntNr": "ACC123456789",
  "GITaxCrAcntNr": "ACC123456789"
},
{
  "TaxTypCd": "TAX04",
```

```

    "TaxLocAmt": 1.00,
    "GITaxDbAcntNr": "ACC123456789",
    "GITaxCrAcntNr": "ACC123456789"
  },
},

```

4.3.2.6 Special characters to be escaped

The following characters are reserved characters and cannot be used in JSON and must be properly escaped to be used 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

Each JSON file can be validated by using its own JSON-Schema. The name of this schema is equal to the root tag name followed by the version of the message like:

“AdcApAdjustments.json”

The JSON Schema is derived from the Audit Data Collection Data Model.

```

{
  "$schema": "http://json-schema.org/draft-06/schema#",
  "$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",
    "Message Rev.: 10",
    "Transaction:  AA - Audit Data Collection",
    "Transaction Rev.: 277 - Audit Data Collection",
    "Datamodel:    Audit Data Collection Datamodel, version HEAD",
    "Datamodel Rev.: 378",
    "XML tagset used: AdcsML",
    "Generated at: 29-05-2021 21:17:14",
    "NOTE: This schema was generated automatically: manual modifications will be lost",
    "NOTE: Additional rules/conditions regarding the definition may exist, see functional hierarchical specification"
  ],
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "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"
  },
  "AdjTypNm": {
    "$ref": "../definitions/domains/TypText60",
    "title": "Adjustment Type Name"
  },
  "AdjDocNr": {
    "$ref": "../definitions/domains/TypText100",
    "title": "Adjustment Document Number"
  },
  "InvoId": {
    "$ref": "../definitions/domains/TypId60",
    "title": "Invoice ID"
  },
  "JrnlId": {
    "$ref": "../definitions/domains/TypId100",
    "title": "Journal ID"
  },
  "FiscYr": {
    "$ref": "../definitions/domains/TypYrId",
    "title": "Fiscal Year"
  },
  "AcntingPer": {
    "$ref": "../definitions/domains/TypPerId",
    "title": "Accounting Period"
  },
  "AdjDt": {
    "$ref": "../definitions/domains/TypDt",
    "title": "Adjustment Date"
  },
  "SuplAcntId": {
    "$ref": "../definitions/domains/TypId100",
    "title": "Supplier Account ID"
  },
  "GIDbAcntNr": {
    "$ref": "../definitions/domains/TypGIAcntNr100",
    "title": "GL Debit Account Number"
  },
  "GICrAcntNr": {
    "$ref": "../definitions/domains/TypGIAcntNr100",
    "title": "GL Credit Account Number"
  },
  "TrAmt": {
    "type": "object",
    "additionalProperties": false,
    "properties": {
      "FuncAmt": {
        "$ref": "../definitions/domains/TypAmt",
        "title": "Functional Amount"
      },
      "FuncCurCd": {
        "$ref": "../definitions/domains/TypCurCd",

```

```

        "title": "Functional Currency Code"
    },
    "TrAmt": {
        "$ref": "../definitions/domains/TypAmt",
        "title": "Transaction Amount"
    },
    "TrCurCd": {
        "$ref": "../definitions/domains/TypCurCd",
        "title": "Transaction Currency Code"
    },
    "RprtAmt": {
        "$ref": "../definitions/domains/TypAmt",
        "title": "Reporting Amount"
    },
    "RprtCurCd": {
        "$ref": "../definitions/domains/TypCurCd",
        "title": "Reporting Currency Code"
    },
    "LocAmt": {
        "$ref": "../definitions/domains/TypAmt",
        "title": "Local Amount"
    },
    "LocCurCd": {
        "$ref": "../definitions/domains/TypCurCd",
        "title": "Local Currency Code"
    }
},
"required": []
},
"Tax": {
    "type": "array",
    "title": "ap adjustment - TAX",
    "minItems": 0,
    "maxItems": 4,
    "items": {
        "type": "object",
        "additionalProperties": false,
        "properties": {
            "TaxTypCd": {
                "$ref": "../definitions/domains/TypTaxTypCd",
                "title": "Tax Type Code"
            },
            "TaxLocAmt": {
                "$ref": "../definitions/domains/TypAmt",
                "title": "Tax Local Amount"
            },
            "GltaxDbAcntNr": {
                "$ref": "../definitions/domains/TypGltaxAcntNr100",
                "title": "GL Tax Debit Account Number"
            },
            "GltaxCrAcntNr": {
                "$ref": "../definitions/domains/TypGltaxAcntNr100",
                "title": "GL Tax Credit Account Number"
            }
        }
    },
    "required": []
}
}

```



```

},
"Crea": {
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "UserId": {
      "$ref": "#/definitions/domains/TypId25",
      "title": "User ID"
    },
    "Dt": {
      "$ref": "#/definitions/domains/TypDt",
      "title": "Date"
    },
    "Tm": {
      "$ref": "#/definitions/domains/TypTm",
      "title": "Time"
    }
  },
  "required": []
},
"Aprv": {
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "UserId": {
      "$ref": "#/definitions/domains/TypId25",
      "title": "User ID"
    },
    "Dt": {
      "$ref": "#/definitions/domains/TypDt",
      "title": "Date"
    },
    "Tm": {
      "$ref": "#/definitions/domains/TypTm",
      "title": "Time"
    }
  },
  "required": []
},
"LstMdf": {
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "UserId": {
      "$ref": "#/definitions/domains/TypId25",
      "title": "User ID"
    },
    "Dt": {
      "$ref": "#/definitions/domains/TypDt",
      "title": "Date"
    },
    "Tm": {
      "$ref": "#/definitions/domains/TypTm",
      "title": "Time"
    }
  },
  "required": []
}

```

```

    },
    "BusSg": {
      "type": "array",
      "title": "ap adjustment - BUSINESS SEGMENT",
      "minItems": 0,
      "maxItems": 999999,
      "items": {
        "type": "object",
        "additionalProperties": false,
        "properties": {
          "BusSgCd": {
            "$ref": "#/definitions/domains/TypId25",
            "title": "Business Segment Code"
          }
        }
      },
      "required": []
    }
  },
  "required": [ "AdjId" ]
}
},
"required": [],
"definitions": {
  "domains": {
    "TypAmt": {
      "type": "number",
      "multipleOf": 0.0001,
      "title": "AMOUNT"
    },
    "TypCurCd": {
      "type": "string",
      "minLength": 3,
      "maxLength": 3,
      "title": "CURRENCY CODE"
    },
    "TypDt": {
      "type": "string",
      "minLength": 10,
      "maxLength": 10,
      "pattern": "^\\d{4}-(0[1-9]|1[0-2])-(0[1-9]|1[2]\\d|3[0-1])$",
      "title": "DATE (CCYY-MM-DD)"
    },
    "TypGIAcntNr100": {
      "type": "string",
      "maxLength": 100,
      "title": "GL ACCOUNT NUMBER 100"
    },
    "TypId100": {
      "type": "string",
      "maxLength": 100,
      "title": "IDENTIFIER 100"
    },
    "TypId25": {
      "type": "string",
      "maxLength": 25,

```

```

        "title": "IDENTIFIER 25"
    },
    "TypId60": {
        "type": "string",
        "maxLength": 60,
        "title": "IDENTIFIER 60"
    },
    "TypPerId": {
        "type": "string",
        "maxLength": 15,
        "title": "PERIOD IDENTIFIER"
    },
    "TypTaxTypCd": {
        "type": "string",
        "maxLength": 25,
        "title": "TAX TYPE CODE"
    },
    "TypText100": {
        "type": "string",
        "maxLength": 100,
        "title": "TEXT 100"
    },
    "TypText60": {
        "type": "string",
        "maxLength": 60,
        "title": "TEXT 60"
    },
    "TypTm": {
        "type": "string",
        "maxLength": 8,
        "title": "TIME"
    },
    "TypYrId": {
        "type": "integer",
        "title": "YEAR IDENTIFIER"
    }
}
}
}

```

An overview of the existing JSON schemas can be found in Annex C

4.3.4 JSON Sample message

In the following example there is one table with one line in one JSON file.

Remark: In this stage of the project it was still not decided to use a root tag or name space in a JSON instance, this means that the ADCS table full name only can be derived from the file name.

```

{
  "ApAdj": [{
    "AdjId": "AD1234",
    "AdjNr": "123456789",
    "AdjTypNm": "Finance charge.",
    "AdjDocNr": "123456789",
    "InvoId": "I001",
    "JrId": "JRN1",
    "FiscYr": 2020,
  }
]
}

```

```

"AcntingPer": "M1",
"AdjDt": "2020-02-01",
"SuplAcntId": "S00001",
"GIDbAcntNr": "ACC123456789",
"GICrAcntNr": "ACC123456789",
"TrAmt": {
  "FuncAmt": 125.1255,
  "FuncCurCd": "USD",
  "TrAmt": 125.1255,
  "TrCurCd": "USD",
  "RprtAmt": 125.1255,
  "RprtCurCd": "USD",
  "LocAmt": 125.1255,
  "LocCurCd": "USD"
},
"Tax": [{
  "TaxTypCd": "TAX01",
  "TaxLocAmt": 12.50,
  "GITaxDbAcntNr": "ACC123456789",
  "GITaxCrAcntNr": "ACC123456789"
},
{
  "TaxTypCd": "TAX02",
  "TaxLocAmt": 5.00,
  "GITaxDbAcntNr": "ACC123456789",
  "GITaxCrAcntNr": "ACC123456789"
},
{
  "TaxTypCd": "TAX03",
  "TaxLocAmt": 0.75,
  "GITaxDbAcntNr": "ACC123456789",
  "GITaxCrAcntNr": "ACC123456789"
},
{
  "TaxTypCd": "TAX04",
  "TaxLocAmt": 1.00,
  "GITaxDbAcntNr": "ACC123456789",
  "GITaxCrAcntNr": "ACC123456789"
}
}],
"Crea": {
  "UserId": "JD",
  "Dt": "2020-03-02",
  "Tm": "09:31"
},
"Aprv": {
  "UserId": "JD",
  "Dt": "2020-03-02",
  "Tm": "09:31"
},
"LstMdf": {
  "UserId": "JD",
  "Dt": "2020-03-02",
  "Tm": "09:31"
},
"BusSg": [{
  "BusSgCd": "1000"
},

```

```
{
  "BusSgCd": "1200"
},
{
  "BusSgCd": "1234"
}]
}]
}
```

4.4 CSV

4.4.1 General

To keep the three exchange formats (XML, JSON, CSV) consistent, it is also explained 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 will be possible. This technical guideline for CSV is recommended, but optional. If users do not need conversion to XML or JSON, it's allowed to just follow ISO 21378:2019 base specification.

4.4.2 Technical guideline

A comma-separated values (CSV) file, with the filename ending in .csv, stores tabular data, including numbers and text, in plain text. In the CSV file, each line of the file is a data record, 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 repeating lines, must be sent in one CSV 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>
```

Italics printed text should 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> should be replaced by the line break characters used on the target operating system (e.g., CR/LF on Windows and LF on Unix systems).

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

If CR/LF occurs in the data, it will not be escaped as textual data is enclosed by double quotes.

4.4.2.1 Construction of column header tags

CSV files use the XML tags as column header tags.

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

4.4.2.2 Representation of (repeating) groups

Column tags of data elements which belong to a group will have the group tag followed by a dot (.) and followed by their own tag. For instance, "FuncAmt" in group "TrAmt" will have column header tag "TrAmt.FuncAmt".

Column header tags of data elements within a repeating group are prefixed by the group tag plus a number between () from 1 to the maximum number of repeats and then 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. until 4.

4.4.2.3 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.

4.4.2.4 Special characters to be escaped

Quotation mark ("). If this character occurs in the data, it should be doubled like "".

4.4.2.5 Optional fields

Although certain fields or columns may not be required, in a CSV file the columns with their column header must always be present. For an optional field that is not present we will see two adjacent column separators in the CSV file (e.g. ;).

4.4.3 CSV validation

For CSV validation there exists a Schema language:

[CSV Schema Language 1.1 \(digital-preservation.github.io\)](https://digital-preservation.github.io/CSV-Schema-Language-1.1/)

However this is an unofficial draft.

4.4.4 CSV Sample

In the following example in text format, there's one table with one header line and one data content line in one CSV file:

```
AdjId;AdjNr;AdjTypNm;AdjDocNr;InvoiId;JrnId;FiscYr;AcntingPer;AdjDt;SuplAcntId;GlDbAcntNr;GlCrAcntNr;TrAmt.FuncAmt;TrAmt.FuncCurCd;TrAmt.TrAmt;TrAmt.TrCurCd;TrAmt.RprtAmt;TrAmt.RprtCurCd;TrAmt.LocAmt;TrAmt.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;ACC123456789;ACC123456789;125.1255;USD;125.1255;USD;125.1255;USD;TAX01;12.50;ACC123456789;ACC123456789;TAX02;5.00;ACC123456789;ACC123456789;TAX03;0.75;ACC123456789;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
```

The above CSV example opened in OpenOffice Calc:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	AdjId	AdjNr	AdjTypNm	AdjDocNr	InvoiId	JrnId	FiscYr	AcntingPer	AdjDt	SuplAcntId	GlDbAcntNr	GlCrAcntNr	TrAmt.FuncAmt	TrAmt.FuncCurCd
2	AD1234	123456789	Finance charge.	123456789	I001	JRN1	2020	M1	2020-02-01	S00001	ACC123456789	ACC123456789	125.1255	USD
3														
4														
5														
6														

Figure 1 --- CSV sample presented by OpenOffice Calc

4.5 Mapping ISO 21378:2019 ADCS Tables to XML and JSON

This chapter specifies the mapping between ISO 21378:2019 ADCS Tables and XML or JSON or vice versa. One mapping table per ISO 21378:2019 ADCS Table must be set up. So, 71 mapping tables must be created.

For each data element in ISO 21378:2019 ADCS it is defined which element (or combined elements) in XML and JSON is to be used to contain its information contents.

The next chapter shows an example of such a table.

4.5.1 Mapping Table sample

Mapping ISO 21378:2019 ADCS table to XML or JSON and vice versa.

Table : AP_Adjustments
Version: : 1.0
XML-Schema : AdcApAdjustments.xsd
Namespace : <http://schemas.iso.org/AdcsML/Messages/AdcApAdjustments-v1>
JSON-Schema : AdcApAdjustments_jsd6.json
Roottag : AdcApAdjustments
Table tag : ApAdj
Path : AdcApAdjustments.ApAdj."XML / JSON Tag"

No.	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	InvoId
6	Journal_ID	String	JrnId
7	Fiscal_Year	String	FiscYr
8	Accounting_Period	String	AcntingPer
9	Adjustment_Date	Date	AdjDt
10	Supplier_Account_ID	String	SuplAcntId
11	Adjustment_Functional_Amount	Decimal	TrAmt.FuncAmt
12	ADJ_Functional_CUR_Code	String	TrAmt.FuncCurCd
13	ADJ_Transaction_Amount	Decimal	TrAmt.TrAmt
14	ADJ_TRX_CUR_Code	String	TrAmt.TrCurCd
15	Adjustment_Reporting_Amount	Decimal	TrAmt.RprtAmt
16	Adjustment_Reporting_CUR_Code	String	TrAmt.RprtCurCd
17	Adjustment_Local_Amount	Decimal	TrAmt.LocAmt
18	Adjustment_Local_Currency_Code	String	TrAmt.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
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).BusSgCd n = 1,2,3 etc.

4.6 Cross topics

4.6.1 General

In addition to the technical part, different data formats face the same problems, which are management level, such as encryption of sensitive data.

4.6.2 Character encoding conventions

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

ASCII may be used in English, Chinese national standard 18030 may be used in China.

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

4.6.3 Versioning

Versioning has two primary aspects.

- a) Versioning of the definitions: Because the ADCS may 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 the audit data collection, 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 are complete. This will help mitigate errors, omissions or other issues that may require re-extracting the data. It is therefore important to be able to easily differentiate between files extracted using the different versions of the standard. In addition, it is also important to have the ability to distinguish later extracts that replace or augment an original extract.

Over time, the standard may need to be refreshed to capture the types of changes referenced above. 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, some suggestions are as follows.

- 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. Three-digit Arabic numerals should be added at the end of the filename to express the file order.

EXAMPLE xxx_001.xml and xxx_002.xml.

- 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 may corrupt the file and render it unreadable.

4.6.5 Limiting file size

Limiting file sizes can improve the data transfer efficiency.

When the file exceeds a maximum size (for instance 100,000 lines) the file must be split into several files, each not exceeding the maximum number of lines. The sequence number of the file is expressed in the file name (extracted table piece count), see chapter 4.6.7.

Remark: At this stage of the project the maximum number of lines was not yet decided for XML and JSON files.

4.6.6 File Compression

File compressing means representing a source file with less bits, which can improve the data transfer efficiency and reduce storage consumption and is useful in data transmission and storage.

For XML and JSON files it is highly recommended to compress all files and packed in a folder. In general, a compressed XML file is only 10% of its uncompressed size.

4.6.7 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 must be built into the directory structure and filenames. The folder-naming conventions and some examples are shown in the next table:

Table 1 — Folder-naming conventions and examples

Type of information	Information example	Folder-naming examples
Company	<Widget, Inc>	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 the next table:

Table 2 — 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_00023_001.xml
Period end extracted	<October>, <7>	
Year extracted	<2018>, <19>	
Company	<Widget, Inc>	
Module	<GL>, <INV>	
Table name	<SAL_Orders>, <INV_Location>	
Extraction date	<20181108>	
Extraction unique ID	<EID00001>, <XID1>, <00023>	
Extracted table piece count	<001>, <002>, <003>	
Extension	<xml>,<json>,<csv>,<txt>	

4.6.8 Representation of special field types

Although the field types are defined in the ADCS standard, it is useful to explain some general types in this document:

Format of Date fields: CCYY-MM-DD (e.g. 2020-03-30)

Format of Time fields: HH:MM (e.g. 20:30)

Format of Amounts, Prices, Percentages: Will 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. (e.g. 100.25). Leading zeros are not allowed. For example, "0123.50" is not allowed, which must be written as "123.50"; "00.05" is not allowed and must be written as "0.05".

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

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

4.6.9 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, there is the possibility to extend ADCS with new tables.

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

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

Extensions must 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 may be moved to the original ADCS table in a new version of the standard.

4.6.10 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 audit data collection. Data encryption is an important means to protect private data.

Remark: At this stage of the project, no decision has yet been made on how the XML and JSON files should be encrypted.

Annex A

XML Tags abbreviation List

1	1	Abbreviation	Abrv
1	2	Academic	Acad
1	3	Account	Acnt
1	4	Accrual	Accrl
1	5	Accumulated	Accum
1	6	Acquisition	Acq
1	7	Addition	Add
1	8	Address	Adr
1	9	Adjusted	Adjd
1	10	Adjustment	Adj
1	11	After	Aft
1	12	Allocation	Alloc
1	13	Amount	Amt
1	14	Application	Apl
1	15	Approved	Aprv
1	16	Balance	Bal
1	17	Before	Bef
1	18	Beginning	Beg
1	19	Billing	Bill
1	20	Branch	Bra
1	21	Business	Bus
1	22	Cancellation	Cncl
1	23	Change	Chng
1	24	Code	Cd
1	25	Contact	Cnt
1	26	Content	Cont
1	27	Contract	Contr
1	28	Corresponding	Corr
1	29	Costing	Cstng
1	30	Country	Cntry
1	31	Created	Crea
1	32	Credit	Cr
1	33	Currency	Cur
1	34	Customer	Cust
1	35	Customized	Cstmzd
1	36	Date	Dt
1	37	Debit	Db
1	38	Default	Dft
1	39	Department	Dep
1	40	Depreciable	Dprcbl

1	41	Depreciation	Depre
1	42	Description	Dscr
1	43	Details	Dtls
1	44	Developer	Dvlpr
1	45	Discount	Dscnt
1	46	Dispatch	Disp
1	47	Document	Doc
1	48	Employee	Emp
1	49	Employment	Emplmnt
1	50	Encoding	Enc
1	51	Ending	End
1	52	Exclude	Excl
1	53	Expense	Expns
1	54	External	Ext
1	55	Fiscal	Fisc
1	56	Fiscal	Fscl
1	57	Functional	Func
1	58	Generated	Gen
1	59	Grouping	Grp
1	60	Header	Hdr
1	61	Hierarchy	Hrchy
1	62	Identifier	Id
1	63	Impairment	Impr
1	64	Include	Incl
1	65	Indicator	Ind
1	66	Inventory	Inv
1	67	Invoice	Invoi
1	68	Journal	Jrn
1	69	Last	Lst
1	70	Line	Ln
1	71	Local	Loc
1	72	Location	Lct
1	73	Materials	Mat
1	74	Measurement	Mea
1	75	Method	Mthd
1	76	Modified	Mdf
1	77	Module	Mod
1	78	Name	Nm
1	79	Number	Nr
1	80	Open	Opn
1	81	Order	Ord
1	82	Organization	Org
1	83	Parent	Par
1	84	Payable	Pbl
1	85	Payment	Pay
1	86	Percentage	Perc

1	87	Period	Per
1	88	Person	Prsn
1	89	Physical	Phys
1	90	Primary	Prim
1	91	Proceeds	Prcls
1	92	Process	Proc
1	93	Product	Prdct
1	94	Project	Proj
1	95	Proportion	Prop
1	96	Province	Prvnc
1	97	Provision	Prov
1	98	Purchase	Pur
1	99	Purchasing	Prchsng
1	100	Quantity	Qt
1	101	Realized	Rlzd
1	102	Receipt	Rcpt
1	103	Receivable	Rcvbl
1	104	Received	Rcvd
1	105	Records	Rec
1	106	Reference	Ref
1	107	Regulator	Rgltr
1	108	Remaining	Rmng
1	109	Remark	Rmrk
1	110	Removal	Rmv
1	111	Replacemant	Replac
1	112	Replacement	Rplc
1	113	Reporting	Rprt
1	114	Requisition	Rqstn
1	115	Residual	Resi
1	116	Responsibility	Resp
1	117	Reversal	Rev
1	118	Sales	Sal
1	119	Segment	Sg
1	120	Service	Srvc
1	121	Settlement	Setl
1	122	Sheet	Sht
1	123	Shipment	Shp
1	124	Shipping	Shpng
1	125	Software	Sftw
1	126	Source	Src
1	127	Standard	Std
1	128	Status	Stat
1	129	Stocking	Stck
1	130	Subledger	Sbldgr
1	131	Supplier	Supl
1	132	System	Syst

1	133	Terms	Trms
1	134	Time	Tm
1	135	Total	Tot
1	136	Transaction	Tr
1	137	Type	Typ
1	138	Unrealized	Unrlzd
1	139	Version	Vers
1	140	Year	Yr

Annex B

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

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

Data Model

To design the XML schemas and JSON schemas in a consistent and structured way, a data model was created from the Audit Data Collection ISO 21378:2019 according to ISO 11179.

This data model is not part of the ISO 21378:2019 standard but is a means for designing consistent and common structured schemas.

This data model consists of 3 parts:

1. Object Model,
2. Transaction Model based on the Object Model,
3. Message Models based on the Transaction Model and the basis for exchange formats.

(1) Object Model

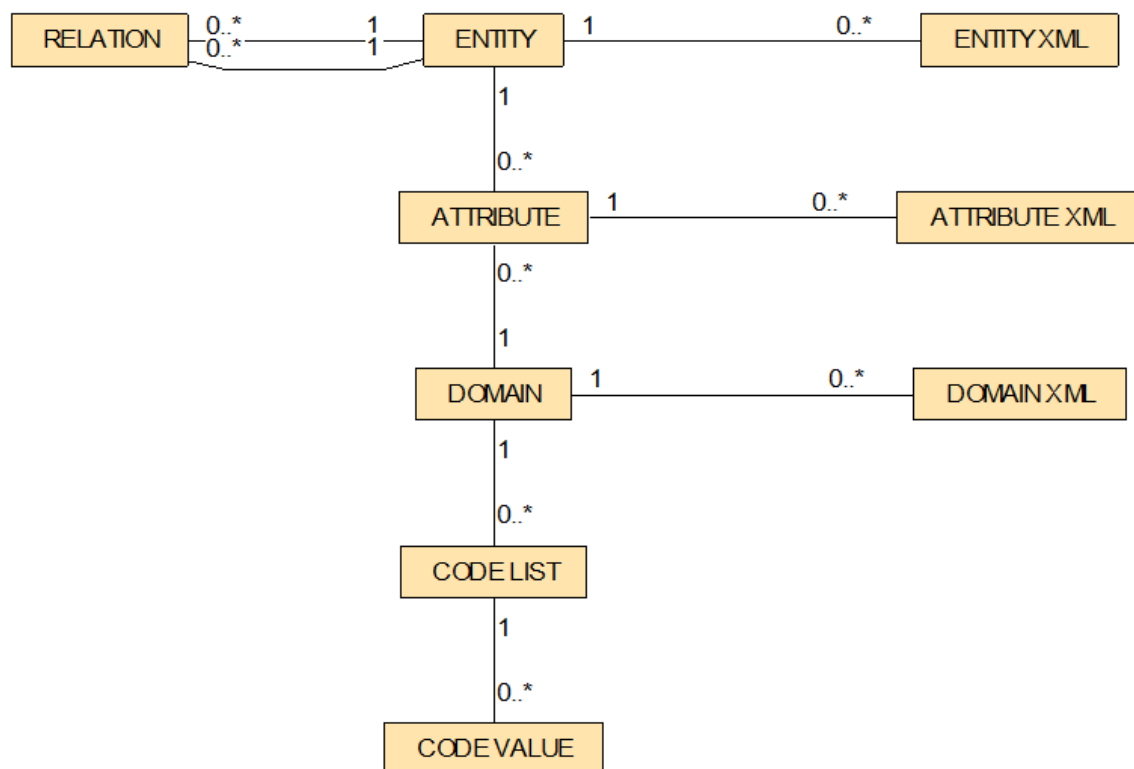


Figure D1 -- Meta data model diagram for the object model

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

Description of the different components in the Object Model:

Entity

In an entity, related data is recorded. In this way, an entity describes an object. An entity has a name and a description.

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

An entity may 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, with the subtypes also inheriting the attributes of the super-type. The subtype has a so-called “is_a relationship” with the super-type.

Relation

A relation is the relationship property between two entities.

Attribute

An attribute describes one property of a single entity. An attribute describes the following:

- a. Name
- b. Definition
- c. Domain reference
- d. Entity reference

Examples of attributes are “Fiscal Year”, “GL Account Number”, “Tax Percentage”.

Domain

A domain describes a class of values with a common scope and structure. A domain has a name, a description and format specifications. A domain can refer to multiple attributes from different entities.

Aspects of domain formats:

- a. Numeric, alphanumeric or alphabetical;
- b. Variable length or fixed length;
- c. Length of the attribute, including + or – sign and decimal point;
- d. Datatype (Amount, Percentage, Date, Time, Text, Boolean etc.)

The chosen datatype is reflected in the XML Schema and JSON Schema.

Examples of domains are AMOUNT, DATE, TIME, TEXT, BOOLEAN

Code-list and Code-value

A code list is a set of values with a common scope. A code list has a name, a responsible agency, and values (codes). A code list is always associated with a single domain.

Entity XML, Attribute XML, Domain XML

Contains the XML specific properties needed to produce XML schemas from a Message Model. The same is also present for JSON and CSV.

ADCS tables modeled in the Object Model:

For all ADCS tables, the following (71) parent entities are defined, which are (in alphabetic order):

AP ADJUSTMENT
AP ADJUSTMENT DETAILS
AP CASH APPLICATION
AP OPEN ACCOUNT PAYABLE
AP PAYMENT MADE
AR ADJUSTMENT
AR ADJUSTMENT DETAILS
AR CASH APPLICATION
AR CASH RECEIVED
AR OPEN ACCOUNT RECEIVABLE
BAS ACCOUNTING PERIOD
BAS BANK ACCOUNT
BAS BILL TYPE
BAS BUSINESS SEGMENT
BAS BUSINESS SEGMENT HIERARCHY
BAS CHART OF ACCOUNTS
BAS CURRENCY
BAS CUSTOMER
BAS CUSTOMER TYPE
BAS CUSTOMIZED ACC SEGMENT
BAS CUSTOMIZED ACC VALUE
BAS EMPLOYEE
BAS JOURNAL ENTRY TYPE
BAS MEASUREMENT UNIT
BAS PAYMENT TERM
BAS PROFILE
BAS PROJECT
BAS SETTLEMENT METHOD
BAS SUPPLIER
BAS SUPPLIER TYPE
BAS TAX REGULATORY
BAS TAX TYPE
BAS USER
GL ACCOUNT PERIOD BALANCE
GL ACCOUNT SEGMENT
GL DETAILS
GL SOURCE
GL TRIAL BALANCE
INV LOCATION
INV ON HAND
INV PERIOD BALANCE
INV PHYSICAL INVENTORY
INV PRODUCT
INV PRODUCT TYPE
INV TRANSACTION
PPE ADDITION
PPE CHANGE
PPE DEPARTMENT ALLOCATION
PPE DEPRECIATION
PPE DEPRECIATION METHOD
PPE MASTER
PPE REMOVAL
PPE TYPE
PUR CONTRACT
PUR CONTRACT DETAILS
PUR INVOICE RECEIVED

PUR INVOICE RECEIVED DETAILS
PUR MATERIALS RECEIVED
PUR MATERIALS RECEIVED DETAILS
PUR ORDER
PUR ORDER DETAILS
PUR REQUISITION
PUR REQUISITION DETAILS
SAL CONTRACT
SAL CONTRACT DETAILS
SAL INVOICE GENERATED
SAL INVOICE GENERATED DETAILS
SAL ORDER
SAL ORDER DETAILS
SAL SHIPMENT MADE
SAL SHIPMENT MADE DETAILS

The following child entities are defined (in alphabetic order):

ACCOUNT SEGMENT
APPROVED
BALANCE AMOUNT
BALANCE BEGINNING AMOUNT
BALANCE ENDING AMOUNT
BILLING ADDRESS
BUSINESS SEGMENT
CREATED
CREDIT AMOUNT
CURRENCY
DEBIT AMOUNT
LAST MODIFIED
PHYSICAL ADDRESS
POSTED
PRIMARY CONTACT
TAX
TRANSACTION AMOUNT

The following super entities are defined (in alphabetic order)

ACTIVITY
ADDRESS
CONTACT
CONTRACT
CONTRACT DETAILS
INVOICE
INVOICE DETAILS
MONETARY AMOUNTS
ORDER
ORDER DETAILS

Example: AP ADJUSTMENT

Entity Diagram of AP ADJUSTMENT with its parent entity, child entities and super entities:

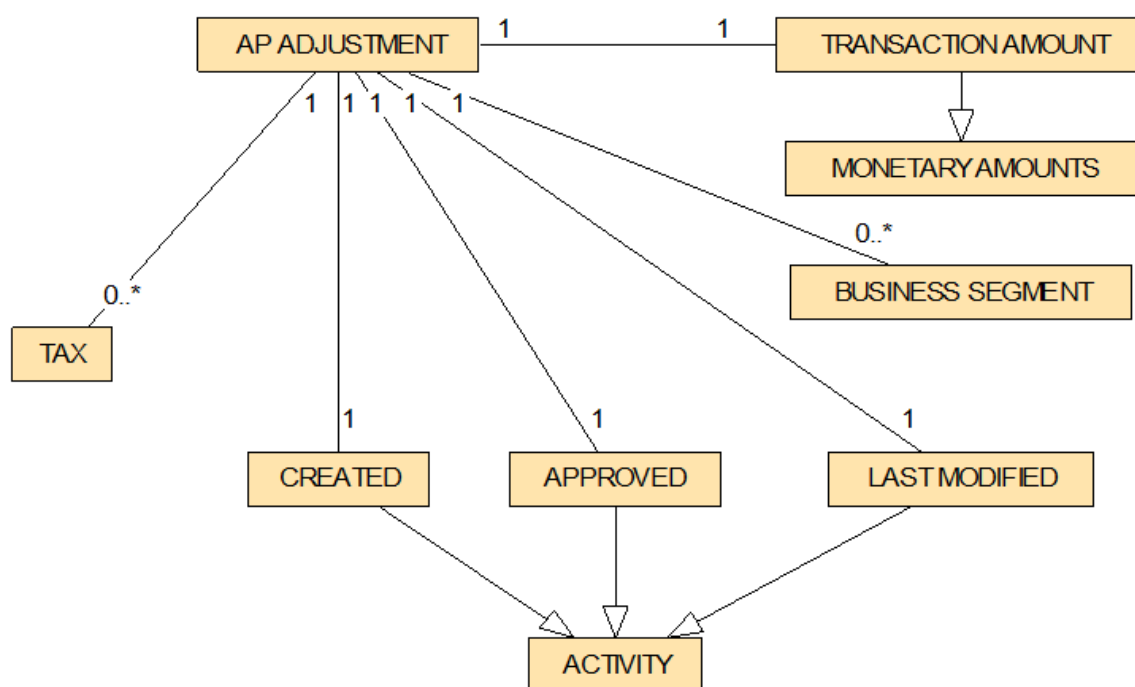


Figure D2 --- AP ADJUSTMENT entity relations diagram

(2) Transaction Model

In order to specify a hierarchical structure for messages, the relevant entities from the object model are displayed in a hierarchical structure. In the Transaction Model, all parent entities are displayed at hierarchical level 1 with the relevant child entities nested below at level 2.

Within the Transaction Model, the relevant attributes from entities can be selected. Only selected attributes will be presented in the messages and schemas.

The following diagram presents the parent entity AP ADJUSTMENT with its child entities in a hierarchic structure, which is the structure for the XML-Schema and JSON-Schema:

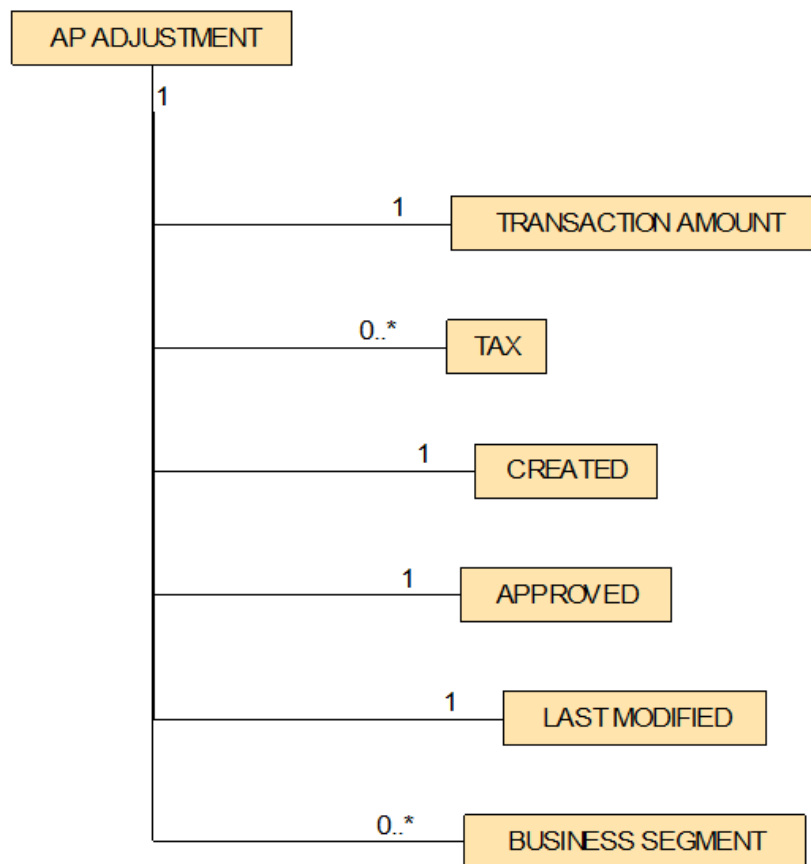


Figure D3 --- AP ADJUSTMENT hierarchic

(3) Message Models

Message models are derived from the transaction model, and therefore have the same hierarchical structure as the transaction model, but the content of the message models will differ, because they are subsets of the transaction model.

A message model includes the following:

1. The name and version of the message,
2. Relevant entities presented in a hierarchical message structure.

An entity within the message model describes the following:

1. Name of entity,
2. Description of the entity,
3. Mandatory or optional indication,
4. Maximum number of times an entity may appear in a message,
5. Relevant attributes.

An attribute describes the following:

1. Name of the attribute,
2. Description of the attribute,
3. Size of the attribute,
4. Mandatory or optional indication,
4. Code list name if an attribute is associated with a code list,
5. Allowed values if an attribute is associated with a code list.

Technical specifications:

Different technical specifications can be derived from a Message Model. For ADCS, these are XML-Schemas, JSON-Schemas and CSV technical specifications.

Data Model overview:

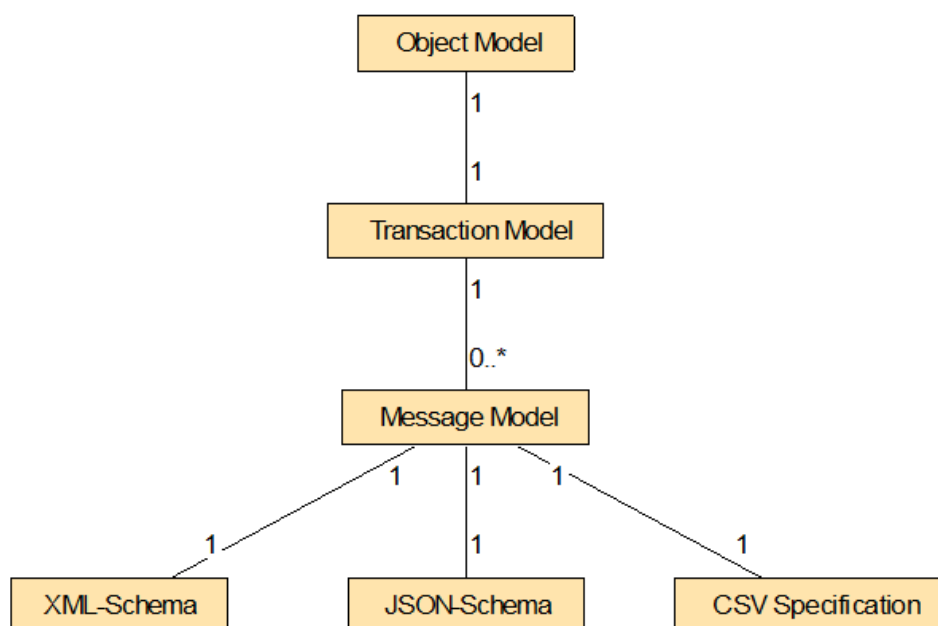


Figure D4 --- Data Model overview

Bibliography

- [1]. RFC 4180, Common Format and MIME Type for Comma-Separated Values (CSV) Files