ECLASS Technical Report 22

Embedding of ECLASS into ISO 29002-10

Version 1.0

01.03.2023

ECLASS Center of Research and Development (CRD):

Ms. Petra Eder

Mr. Oliver Hadasch

Dr. Wilfried Hartmann

Mr. Philippe Juhel

Mr. Gerald Lobermeier

Mr. Stefan Mülhens

Mr. Nikolaus Ondracek

Mr. Frank Scherenschlich

Mr. Josef Schmelter

Mr. Thorsten Kroke

and

Dr. Matthias Richter

Fabian Schwanitz

Yasmin Hassankhani

Please send remarks to crd@eclass.de

Revision History

Date	Version	Change	Who?
08.01.2019	0.1	Initial Draft	RiM
21.01.2019	0.2	Merged with Comments from Wilfried Hartmann	WH/RiM
18.04.2019	0.3	Reworked acc. input from ECLASS CRD	RiM
02.07.2019	1.0	Published as Technical Spec. 22	CH
26.01.2022	1.0	Change the document to current ECLASS template	Hassankhani, Schwanitz
01.03.2023	1.0	Withdrawal of Technical Specification 22 (Version 1) and republish as Technical Report 22 (Version 1)	Schwanitz, Hassankhani

Table of contents

1 Introduction

1.1 Motivation

ECLASS defines a standard for product classification and product description. ECLASS users therefore have the typical use case to exchange data based on releases of the ECLASS Standard in data exchange formats such as BMEcat, GS1 CIN or ISO 29002-10 technical catalog. Since embedding data in a data exchange format usually leaves unclarities or multiple options, ECLASS is handing out guidelines on how to embed payloads based on the ECLASS Standard into particular file formats.

1.2 Purpose

In this document the embedding and usage of ECLASS based content in the context of ISO 29002-10 technical catalogs is described as a guideline to implementers.

This document uses as its schema reference the amended version of ISO 29002-10 catalog that is incorporated in ECLASS XML 3.0 or higher.

1.3 Scope

In scope is:

- Usage of ECLASS XML 3.0 Schema in respect to the encoding of product descriptions according to a release of the ECLASS dictionary
- Usage of ECLASS XML 3.0 Schema in respect to the encoding of product descriptions in ECLASS workflow responses according to a release of the ECLASS dictionary

This document does not describe:

- Handling of dictionary releases or updates
- Conversion of product descriptions from or into other formats
- Conversion from advanced representation into basic representation or vice versa
- Topics from ECLASS workflow that go beyond the encoding of product descriptions

1.4 Definitions, Acronyms and Abbreviations

Technical Catalog a collection of items encoded into ISO 29002-10 XML

Item	a product description encoded into ISO 29002-10 XML

General definitions, acronyms, and abbreviations are described in the ECLASS Technical Support.

1.5 References

ECLASS dictionary Schema

https://eclass.eu/fileadmin/Redaktion/pdf-Dateien/Wiki/ECLASSXML_3.0.1/eClassXML/dictionary.xsd

ECLASS technical catalog Schema

https://eclass.eu/fileadmin/Redaktion/pdf-Dateien/Wiki/ECLASSXML_3.0.1/ontoML/ISO29002/catalogue.xsd

ECLASS Release

More information on ECLASS can be found at: www.eclass.de

ISO 29002-5

ISO/TS 29002-5:2009

Industrial automation systems and integration -- Exchange of characteristic data -- Part 5: Identification scheme

www.iso.org/standard/50773.html

ISO 29002-10

ISO/TS 29002-10:2009

Industrial automation systems and integration -- Exchange of characteristic data -- Part 10: Characteristic data exchange format

www.iso.org/standard/50774.html

1.6 Basic Annotations

The following assumptions are made in this document:

- An ECLASS Release is available in ECLASS Advanced XML format to the user or system that is encoding or decoding the technical catalog
- Concept identifiers are used to identify dictionary elements

2 Prerequisites

2.1 Identification

2.1.1 Concept Identifier (ISO 29002-5)

When referencing ECLASS dictionary elements the entities need to be identified. Since worldwide, multilingual exchange of data by machines is in scope, the identification is not done in natural language dependent names but instead the identification schema described in ISO 29002-5 is used and all references are made by IRDIs (see for its Syntax: IRDI)

Table 1 lists the CSIs relevant for technical catalogs, items and referenced dictionary elements:

Table 1: Selection of Code Space Identifiers

Code Space Identifier (CSI)	Category
01	Class
02	Property
05	Unit of Measurement
07	Property Value
11	Dictionary Release
Z1	Item
Z8	Technical Catalog

2.1.2 Property Path

Often a property is not directly assigned to an application class, but has a context consisting of an aspect, reference properties (and thereby blocks), polymorphic block selections or cardinality counters. This assignment context is represented by a so-called property path, which is a concatenation of IRDIs, special separator characters that denote the type of context and cardinality counters.

The property path does not explicitly contain the application class (AC) as this is already contained in cat:item/@class_ref.

Table 2 lists the separator characters used in property paths:

Table 2: Separator Characters and their Meaning in Property Paths

Separator Character	Meaning
/	Separator between path elements (typically reference properties).
:	At beginning of path: reference to aspect class. After a reference property: reference to selected polymorphic block.
*****	Separator between reference property and cardinality counter.
.	Separator between (level type) property and level indicator (MIN, MAX, NOM, TYP). Can only occur at leaf level.

Note: non-polymorphic block references are omitted from property path since they can be derived from the reference property.

Table 3 gives examples of path syntax for a better understanding:

Table 3: Examples of Path Syntax

Path Syntax	Meaning
0173-1#02-EXA123#001	Property EXA123 directly assigned to the AC.
0173-1#02-EXA124#001/0173-1#02-EXA125#001	Property EXA125 in a block assigned to the AC via reference property EXA124.
0173-1#02-EXA126#001*2/0173-1#02-EXA127#001	Property EXA127 in a block assigned to the AC via reference property EXA126. The block is in pattern or cardinality and the path points to the property in the second block instance.
0173-1#02-EXA128#001:0173-1#01-EXA129#001/0173-1#02-EXA130#001	Property EXA130 in a block assigned to the AC via reference property EXA128. The block is polymorphic and the path points to the property in selected block option EXA129.
0173-1#01-EXA131#001/0173-1#02-EXA132#001	property EXA132 assigned to the AC via its aspect EXA131.

2.1.3 ECLASS Coded Name

The eight-digit ECLASS class code (coded name) is not explicitly contained in the items of the ISO 29002-10 technical catalog, but an item must reference the correct classification class using the classification_ref element (see Section 3.4).

See further: Classification Class – Class code (coded name)

2.2 Delivery of the ECLASS Dictionary

Since items reference only their application class, only the advanced XML representation of the ECLASS dictionary can be used with ISO 29002-10.

2.3 Content Representation

The ISO 29002-10 technical catalog can contain property valuations for simple and complex class structures, therefore the expression potential covers the basic and the advanced ECLASS model. The remainder of this document handles the XML implementation of all requirements from the advanced ECLASS model without making a distinction.

3 ECLASS to ISO 29002-10 Mapping

3.1 Overall Structure

3.1.1 Plain ISO 29002-10 Technical Catalog

Each XML file contains one technical catalog with 1..n items.

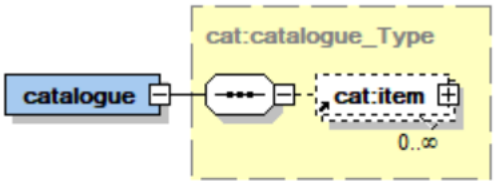

Figure 1: Overall Structure of ISO 29002-10 Catalog

This will produce the following XML:

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

<cat:catalogue

       xmlns:cat="urn:iso:std:iso:ts:29002:-10:ed-1:tech:xml-schema:catalogue"

       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

       xmlns:bas="urn:iso:std:iso:ts:29002:-4:ed-1:tech:xml-schema:basic"

       xmlns:id="urn:iso:std:iso:ts:29002:-5:ed-1:tech:xml-schema:identifier"

       xmlns:val="urn:iso:std:iso:ts:29002:-10:ed-1:tech:xml-schema:value"
```

```
      xsi:schemaLocation="urn:iso:std:iso:ts:29002:-10:ed-1:tech:xml-schema:catalogue
https://www.eclass.eu/static/eClassXML/3.0/ontoML/ISO29002/catalogue.xsd

            urn:iso:std:iso:ts:29002:-10:ed-1:tech:xml-schema:value
https://www.eclass.eu/static/eClassXML/3.0/ontoML/ISO29002/value.xsd

            urn:iso:std:iso:ts:29002:-4:ed-1:tech:xml-schema:basic
https://www.eclass.eu/static/eClassXML/3.0/ontoML/ISO29002/basic.xsd">

<cat:item class_ref="0173-1#01-AAA123#001"

      data_specification_ref="0176-1#11-ECLASS10.1#001"

      local_id="_1">

<!-- property values-->

</cat:item>

<!-- further items -->

</cat:catalogue>
```

0173-1#01-AAA123#001 is the example IRDI of the application class that defines the item.

0176-1#11-ECLASS10.1#001 is the IRDI of ECLASS Release 10.1.

Local IDs need to be unique within the technical catalog.

3.2 Nationalization Conventions for Values

The ISO 29002-10 XML technical catalog format is a machine-oriented format for data exchange, therefore there is no nationalization of its content. This means that decimal numbers are always separated by the dot character (and not the comma).

3.3 Reference to external Files

ISO 29002-10 does not support MIME or other means of referencing external files except by valuating properties of data type URL.

3.4 Reference of ECLASS Release and Classification Class

The ECLASS Release an item adheres to may be made explicit in cat:catalogue / cat:item / @data_specification_ref. Formally this information is not required, since the cat:catalogue /

cat:item / @class_ref already points to a specific version of an application class which is valid for one or more ECLASS Releases.

For being compatible with all future changes regarding decoupling of application class and classification class, the schema optional reference to the classification class is mandatory to be made when embedding ECLASS content. The IRDI of the classification class must be put into cat:catalogue / cat:item / cat:classification_ref.

3.5 Attributes of Item

Each item may have set the attributes listed in Table 4. From schema standpoint only class_ref is mandatory to be set.

Table 4: Attributes of Item

Attribute	Type	Description
class_ref	IRDI	Reference to application class
data_specification_ref	IRDI	Optional reference to dictionary
local_id	String must start with a letter or underscore, and can only contain letters, digits, underscores, hyphens, and periods.	Optional locally unique ID for item
information_supplier_reference_string	String	String by which the information supplier references the item.
is_dependent	Boolean	Can remain unset for ECLASS usage.
is_proprietary	Boolean	When true the data conveyed in the item is meant to be confidential. Can remain unset for ECLASS usage.
is_global_id	Boolean	The supplied ID is also globally unique. Can remain unset for ECLASS usage.
is_model	Boolean	Can remain unset for ECLASS usage.

created_view	IRDI	Can remain unset for ECLASS usage.
view_of	String must start with a letter or underscore, and can only contain letters, digits, underscores, hyphens, and periods.	Reference to a local ID Can remain unset for ECLASS usage.

3.6 Item Reference Numbers

The item may contain any number of cat:reference elements which hold an attribute reference_number and an element designation in the form of translatable text. This is meant to be used to transport reference numbers to the item assigned by suppliers.

3.7 Property Values

3.7.1 Attributes of Property Values

The values of properties are contained in 1..n cat:item/cat:property_value elements. Each of these elements contains the context of the property in the attribute subitem_path_property_ref expressed as a property path (see section 2.1.2). The property reference at the leaf end of the path is repeated in the attribute property_ref. When the optional attribute is_proprietary is true, this indicates whether this property value is meant to be kept confidential.

3.7.2 Value Types of Property Values

The following subsections explain by examples on how the different data types defined ECLASS shall be expressed in an ISO 29002-10 XML technical catalog when they become valuated.

3.7.2.1 Data Type String

A value of data type string is language independent.

```
<cat:property_value
        property_ref="0173-1#02-EXA200#001"
        subitem_path_property_ref="0173-1#02-EXA200#001">
        <val:string_value>DN 50</val:string_value>
</cat:property_value>
```

3.7.2.2 Data Type Translatable

A value of data type string translatable is language dependent.

```
<cat:property_value property_ref="0173-1#02-EXA201#001" subitem_path_property_ref="0173-1#02-EXA201#001">
	<val:localized_text_value>
		<val:content>
			<bas:local_string>
				<bas:content>red</bas:content>
				<bas:language_code>en</bas:language_code>
				<bas:country_code>US</bas:country_code>
			</bas:local_string>
			<bas:local_string>
				<bas:content>rot</bas:content>
				<bas:language_code>de</bas:language_code>
				<bas:country_code>DE</bas:country_code>
			</bas:local_string>
		</val:content>
	</val:localized_text_value>
</cat:property_value>
```

3.7.2.3 Data Type Boolean

In ECLASS, Boolean values have a value list to express the translatable yes / no options instead of Boolean true and false.

```
<cat:property_value property_ref="0173-1#02-EXA202#001" subitem_path_property_ref="0173-1#02-EXA202#001">
	<val:controlled_value value_ref="0173-1#07-CAA017#002" />
</cat:property_value>
```

3.7.2.4 Data Type Integer (count)

```
<cat:property_value property_ref="0173-1#02-EXA203#001" subitem_path_property_ref="0173-1#02-EXA203#001">
```

```
        <val:integer_value>1</val:integer_value>
</cat:property_value>
```

3.7.2.5 Data Type Integer (measure)

```
<cat:property_value property_ref="0173-1#02-EXA204#001" subitem_path_property_ref="0173-1#02-EXA204#001">
        <val:measure_qualified_number_value UOM_ref="0173-1#05-EXA205#001">
                <val:qualified_value>
                        <val:integer_value>1</val:integer_value>
                </val:qualified_value>
        </val:measure_qualified_number_value>
</cat:property_value>
```

3.7.2.6 Data Type Integer (currency)

Currencies can be referenced by currency_code or by currency_ref. Since ECLASS does not maintain currencies, using the currency code from ISO 4217:2015 is appropriate:

```
<cat:property_value property_ref="0173-1#02-EXA206#001" subitem_path_property_ref="0173-1#02-EXA206#001">
        <val:currency_value currency_code="EUR">
                <val:integer_value>1</val:integer_value>
        </val:currency_value>
</cat:property_value>
```

3.7.2.7 Data Type Real (count)

```
<cat:property_value property_ref="0173-1#02-EXA207#001" subitem_path_property_ref="0173-1#02-EXA207#001">
        <val:real_value>1.0</val:real_value>
</cat:property_value>
```

3.7.2.8 Data Type Real (measure)

```
<cat:property_value property_ref="0173-1#02-EXA208#001" subitem_path_property_ref="0173-1#02-EXA208#001">
        <val:measure_qualified_number_value UOM_ref="0173-1#05-EXA209#001">
                <val:qualified_value>
                        <val:real_value>1.0</val:real_value>
```

```
            </val:qualified_value>
      </val:measure_qualified_number_value>
</cat:property_value>
```

3.7.2.9 Data Type Real (currency)

Currencies can be referenced by currency_code or by currency_ref. Since ECLASS does not maintain currencies, using the currency code from ISO 4217:2015 is appropriate:

```
<cat:property_value property_ref="0173-1#02-EXA210#001" subitem_path_property_ref="0173-1#02-EXA210#001">
      <val:currency_value currency_code="EUR">
            <val:real_value>1.0</val:real_value>
      </val:currency_value>
</cat:property_value>
```

3.7.2.10 Data Type Rational (count)

```
<cat:property_value property_ref="0173-1#02-EXA211#001">
      <val:rational_value>
            <val:whole_part>1</val:whole_part>
            <val:numerator>3</val:numerator>
            <val:denominator>4</val:denominator>
      </val:rational_value>
</cat:property_value>
```

3.7.2.11 Data Type Rational (measure)

```
<cat:property_value property_ref="0173-1#02-EXA212#001">
      <val:measure_qualified_number_value UOM_ref="0173-1#05-EXA213#001">
            <val:qualified_value>
                  <val:rational_value>
                        <val:whole_part>1</val:whole_part>
                        <val:numerator>3</val:numerator>
                        <val:denominator>4</val:denominator>
                  </val:rational_value>
            </val:qualified_value>
```

```
      </val:measure_qualified_number_value>
</cat:property_value>
```

3.7.2.12 Data Type Axis 1D

Acc. to ISO 10303 axis1_placement is the direction and location in three-dimensional space of a single axis and is defined by a point (first three numbers) and an axis direction (last three numbers):

```
<cat:property_value property_ref="0173-1#02-EXA214#001" subitem_path_property_ref="0173-1#02-EXA214#001">
      <val:composite_value>
        <val:field>
          <val:real_value>1.0</val:real_value>
        </val:field>
        <val:field>
          <val:real_value>2.0</val:real_value>
        </val:field>
        <val:field>
          <val:real_value>3.0</val:real_value>
        </val:field>
        <val:field>
          <val:real_value>0.0</val:real_value>
        </val:field>
        <val:field>
          <val:real_value>0.0</val:real_value>
        </val:field>
        <val:field>
          <val:real_value>0.0</val:real_value>
        </val:field>
      </val:composite_value>
</cat:property_value>
```

3.7.2.13 Data Type Axis 2D

According to ISO 10303 axis2placement_2d is the location and orientation in two-dimensional space of two mutually perpendicular axes defined by a point (first two numbers) and an axis (last two numbers):

```
<cat:property_value property_ref="0173-1#02-EXA215#001" subitem_path_property_ref="0173-1#02-EXA215#001">
	<val:composite_value>
		<val:field>
			<val:real_value>1.0</val:real_value>
		</val:field>
		<val:field>
			<val:real_value>1.0</val:real_value>
		</val:field>
		<val:field>
			<val:real_value>1.0</val:real_value>
		</val:field>
		<val:field>
			<val:real_value>1.0</val:real_value>
		</val:field>
	</val:composite_value>
</cat:property_value>
```

3.7.2.14 Data Type Axis 3D

According to ISO 10303 axis2placement_3d is the location and orientation in three-dimensional space of two mutually perpendicular axes defined by a point (first three numbers) and two axes (middle and last three numbers):

```
<cat:property_value property_ref="0173-1#02-EXA216#001" subitem_path_property_ref="0173-1#02-EXA216#001">
	<val:composite_value>
		<val:field>
			<val:real_value>3.0</val:real_value>
		</val:field>
```

```
            <val:field>
                  <val:real_value>1.0</val:real_value>
            </val:field>
            <val:field>
                  <val:real_value>1.0</val:real_value>
            </val:field>
            <val:field>
                  <val:real_value>0.0</val:real_value>
            </val:field>
            <val:field>
                  <val:real_value>0.0</val:real_value>
            </val:field>
            <val:field>
                  <val:real_value>1.0</val:real_value>
            </val:field>
            <val:field>
                  <val:real_value>1.0</val:real_value>
            </val:field>
            <val:field>
                  <val:real_value>0.0</val:real_value>
            </val:field>
            <val:field>
                  <val:real_value>0.0</val:real_value>
            </val:field>
      </val:composite_value>
</cat:property_value>
```

3.7.2.15 Data Type Date

```
<cat:property_value property_ref="0173-1#02-EXA217#001" subitem_path_property_ref="0173-1#02-EXA217#001">

      <val:date_value>2013-07-10</val:date_value>
</cat:property_value>
```

3.7.2.16 Data Type Time

```
<cat:property_value property_ref="0173-1#02-EXA218#001" subitem_path_property_ref="0173-1#02-EXA218#001">

      <val:time_value>12:24:00</val:time_value>
</cat:property_value>
```

3.7.2.17 Data Type URL

```
<cat:property_value property_ref="0173-1#02-EXA219#001" subitem_path_property_ref="0173-1#02-EXA219#001">
```

```
	<val:file_value>
		<val:URI>https://www.eclass.de/</val:URI>
	</val:file_value>
</cat:property_value>
```

3.7.2.18 Value from Lists of (coded) Values

```
<cat:property_value property_ref="0173-1#02-EXA220#001" subitem_path_property_ref="0173-1#02-EXA220#001">
	<val:controlled_value value_ref="0173-1#07-EXA221#001" />
</cat:property_value>
```

3.7.3 Environment of Property Values

When the property value depends on a set of conditions (formulated themselves as property values), these can be listed inside the val:environment element of the cat:property_value.

This is currently not modelled in ECLASS dictionary and therefore assumed to remain unset.

3.8 Handling of Level Type

Level type properties are expressed as a measure_qualified_number_value which has 1..4 qualified_value elements inside and a UOM_ref . Each of them can be qualified with one of the four @qualifier_code entries (MIN, MAX, NOM, TYP) and has an integer_value / real_value / rational_value element inside according to the properties' data type:

```
<cat:property_value property_ref="0173-1#02-EXA220#001" subitem_path_property_ref="0173-1#02-EXA220#001">
	<val:measure_qualified_number_value UOM_ref="0173-1#05-EXA221#001">
		<val:qualified_value qualifier_code="MIN">
			<val:real_value>1.0</val:real_value>
		</val:qualified_value>
		<val:qualified_value qualifier_code="MAX">
			<val:real_value>10.0</val:real_value>
		</val:qualified_value>
	</val:measure_qualified_number_value>
</cat:property_value>
```

3.9 Handling of Multi-Valuation

ISO 29002-10 does not define one single solution for multi-valuation but allows a set of containers. ECLASS restricts the containers to be used to the ones described in Table 5.

Table 5: ECLASS supported Types of Multi-Valuation Containers from ISO 29002-10

Type	Meaning
Combination	Any of the given values apply to an individual of the described item
sequence_value	Ordered list of values, with duplicates contained

Example case 1: Confetti in Germany or Austria is sold in packs of all different colors. In this case one might want to multi-valuate the property color to e.g. “red”, “green”, “blue”, “yellow”, “pink”, … and denote that all values of the property apply to the product:

```
<cat:property_value property_ref="0173-1#02-EXA300#001" subitem_path_property_ref="0173-1#02-EXA300#001">
	<val:combination>
		<val:string_value>red</val:string_value>
		<val:string_value>green</val:string_value>
		<val:string_value>blue</val:string_value>
		<val:string_value>yellow</val:string_value>
		<val:string_value>pink</val:string_value>
	</val:combination>
</cat:property_value>
```

(string values used to shorten the example)

Example case 5: When describing a resistor with ring color codes printed on, one could use a sequence_value “brown”, “red”, “brown”, “gold” for a resistor with 120 Ohms and a tolerance of 5%:

```
<cat:property_value property_ref="0173-1#02-EXA304#001" subitem_path_property_ref="0173-1#02-EXA304#001">
	<val:sequence_value>
		<val:string_value>brown</val:string_value>
		<val:string_value>red</val:string_value>
```

```
            <val:string_value>brown</val:string_value>
            <val:string_value>gold</val:string_value>
        </val:sequence_value>
</cat:property_value>
```

(string values used to shorten the example)

Appendix

A File Naming Scheme

File names shall contain:

- Name of the file provider
- Language
- Time stamp

File names shall not exceed 40 characters.

Spaces shall not be used in the file name (please use sublines instead).