GUIDELINE

How to transport ECLASS in the Asset Administration Shell

October 2024

Imprint

Publisher
Industrial Digital Twin Association
Lyoner Strasse 18
60528 Frankfurt am Main
Germany
https://www.industrialdigitaltwin.org/

Version history

Date	Version	Changes made
11.01.2024	0.9	First version with schema version 2 (update needed)
23.08.2024	1.0	Update to schema version 3
10.10.2024	1.0	Release of the Guideline

Contents

Figures

Tables

1 About this Document

This document is the result of the work of the two associations ECLASS and IDTA in order to bring both standards closer together. The content was coordinated in the IDTA workstream "ECLASS Semantics" also with participation of ECLASS members and it is made available in a combined IDTA and ECLASS layout. At ECLASS, this document is listed as a Technical Specification with the same name and the number 28.

This document enables software developers and asset data modellers to transport ECLASS content using the Asset Administration Shell (AAS).

2 General

2.1 Introduction

ECLASS as the leading industrial standard classification system is one of the main semantic toolset used in the Asset Administration Shell (AAS). Bringing both standards closer together two perspectives are relevant in this context:

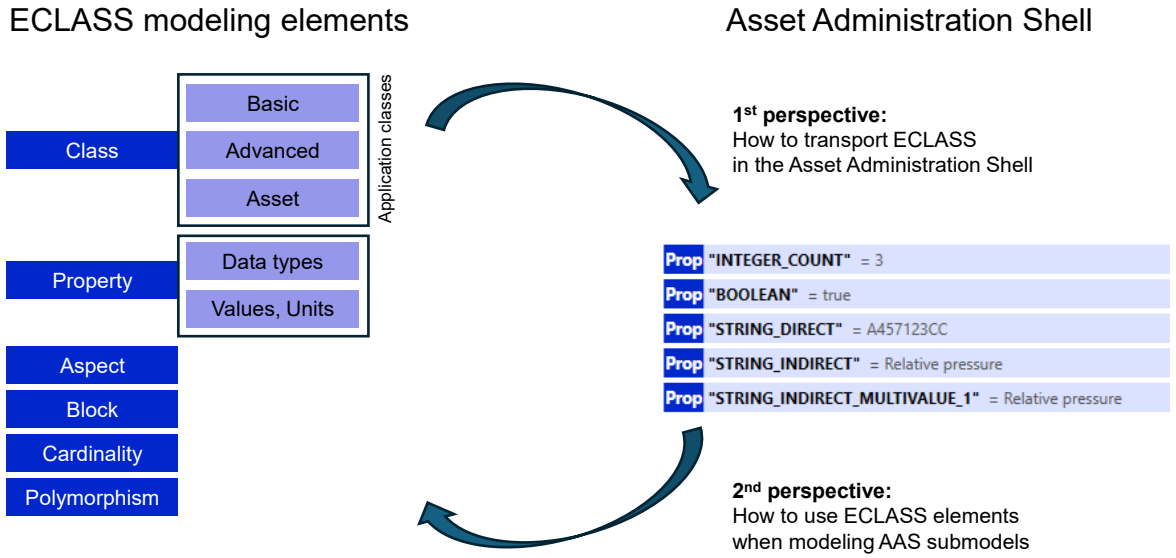


Figure 1: Two perspectives for the combination of ECLASS and AAS

The two perspectives have different relevances:

- **Perspectice 1: HOW to transport ECLASS in the Asset Administration Shell**
  The Asset Administration Shell is also used to transport technical product data based on classifications. Using ECLASS as an example, it is explained in detail how classifications, classification elements up to properties can be transferred.
- **Perspective 2: How to use ECLASS modeling elements when modeling AAS Submodels**
  Classification standards are used to model Submodels. Using ECLASS as an example, it is shown how Submodels can be created using ECLASS.

In this document the first perspective is described in detail.

2.2 ECLASS

The definition and use of ECLASS can be found in http://www.eclass.eu/

All examples based on ECLASS release 14.0, which is available since December 2023.

2.3 Asset Administration Shell (AAS)

More information about the AAS can be found in https://industrialdigitaltwin.org/

This document uses the schema version 3.0.

The Package Explorer version v2024-06-10.alpha is used as a reference application.
https://github.com/eclipse-aaspe/package-explorer/releases/tag/v2024-06-10.alpha

2.4 Related Submodels

This documentation uses the Submodel “Generic frame for Technical Data for industrial equipment in manufacturing” (IDTA 02003-2-0) in version 2.0 (currently in review) as a base. As common this Submodel will named only “Technical Data” in this document.

2.5 Scope / Out of Scope

The scope of this document is the exchange of ECLASS elements in the AAS. That means that ECLASS is used as a semantic in the AAS. Other semantics like IEC CDD can also be used. So, the perspective is based on the ECLASS elements and how they can be transported by using the AAS framework.

3 General definitions

ECLASS modeling elements are used in three representations - ECLASS Basic, ECLASS Advanced and ECLASS Asset (since release 13.0). Depending on the possible field of application, which is described in section 3.1.1, the ECLASS modeling elements are to be transferred using the asset administration shell AAS. Especially in context of ECLASS Advanced the reproducible hierarchy of the ECLASS elements under the application class is necessary, as represented by the ECLASS Technical Specification 15 “URI path”.

3.1 ECLASS elements

In the following list, the relevant ECLASS modeling elements are compared with the corresponding AAS Submodel elements. The associated concept descriptions are considered in the context of the Submodel elements. The section in the document where a more detailed examination takes place is also indicated.

ECLASS element	AAS element	Chapter
Structure	---	4.2
Classification class	---	4.2
Application class	---	4.2
Aspect	Submodel (SM) or SubmodelElementCollection (SMC)	6.1
Property >> details in Table 2	for example: Property (Prop) and MultiLanguageProperty (MLP)	5
Block	SubmodelElementCollection (SMC)	6.2
Cardinality	SubmodelElementList (SML) and SubmodelElementCollection (SMC) in combination with cardinality	6.3
Polymorphism (without Cardinality)	SubmodelElementCollection (SMC)	6.4
Polymorphism (with Cardinality)	SubmodelElementCollection (SMC)	6.5

Table 1: ECLASS elements

In some cases there is no 1:1 relation available between the ECLASS modeling element and the AAS element. For more details, see the following relevant chapters.

3.2 International Registration Data Identifier (IRDI)

The International Registration Data Identifier (IRDI) is based on the international standards ISO/IEC 11179-6, ISO 29002 and ISO 6532 and used in ECLASS and the Asset Administration Shell as unique identifier.

For more information see
https://eclass.eu/support/technical-specification/structure-and-elements/irdi

3.3 URI PATH

This chapter referres to the ECLASS Technical Specification 15 “URI Path” (version 04/2024).

https://eclass.eu/fileadmin/Redaktion/pdf-Dateien/Wiki/ECLASS-Technical-Specification-15-URI-Path-V1.0.pdf

ECLASS Advanced and ECLASS Asset are hierarchical structures based on a nested IRDI structure. To transport ECLASS it is necessary that all relevant IRDI information is available to recreate the correct URI path. That means that a creation of the URI path must be possible with the information that is transported in the AAS model.

There are two general approaches to integrate the information of the URI path into an AAS:

- **absolute**
  Representation of the complete URI path for each element starting with the IRDI of the relevant application class

- **relative (choosen solution)**
  Representation of the relevant elements of the URI path so that the complete URI path can createde using the hierarchy in the AAS

Examples for both approaches are given in the referenced document.

ECLASS Identifiers can be transferred in 2 ways:

- *IRI : https://api.eclass-cdp.com/0173-1-02-AAC895-008*
  - ECLASS offers a webservice where the information can be retrieved
- *IRDI : 0173-1#02-AAC895#008*
  - Direct IRDI that can be used in context of an existing ECLASS dictionary

Both ways are described with an example in chapter 5.1.1. It can be used in all other cases in the same way. In the other examples only the IRI is used. In the corresponding example both identifiers are included for all described cases.

4 Classification

4.1 AAS Submodels

AAS Submodels correspond by definition with an ECLASS Aspect. These ECLASS Aspects can be located only hierarchically direct under the ApplicationClass (ADVANCED, ASSET).

4.2 ECLASS elements in context of classification

The ECLASS classification consist of a 4-level-hierachy with 4 ClassificationClasses followed by three ApplicationClasses (Asset since ECLASS release 13.0). In addition to this, all ClassificationClasses are defined by a ClassCodedName, which is usually known as the ECLASS class number.

For information: The ClassCodedName name will be used in the original way and also here without hypens (27274001 instead of 27-27-40-01).

Structure	Description			
L L L L L L	Element name	ClassCodedName	Class Name	IRDI
●	Segment	27000000	Electric engineering, automation, process control engineering	0173-1#01-AAB572#009
●	Main group	27270000	Sensor technology, safety-related sensor technology	0173-1#01-AFZ578#013
●	Group	27274000	Inductive Sensor	0173-1#01-AHB175#003
●	Class / Subgroup	27274001	Inductive proximity switch	0173-1#01-AGZ376#021
●	ApplicationClass	---	ApplicationClass - Basic	0173-1---BASIC_1_1#01-ABT934#018
●	ApplicationClass	---	ApplicationClass - Advanced	0173-1---ADVANCED_1_1#01-ADN934#013
●	ApplicationClass	---	ApplicationClass - Asset	0173-1---ASSET_1_1#01-AHO549#002

Figure 2: ECLASS classification elements

For information: In the column “IRDI” the “real” IRDI is used. The IRDI notation according to the URI path notation is also available as a primary solution (see chapter 3.3).

For the transport only the last level of the ClassificationClass (named subgroup or class) is relevant.

- For ECLASS Basic and Advanced the corresponding ApplicationClasses are respectively relevant
- For ECLASS Asset the ApplicationClass is a collector of ECLASS Aspects which represent a Submodel

4.3 ECLASS class in Submodel “Technical Data”

ECLASS can be transported using the Submodel “Generic Frame for Technical Data for Industrial Equipment in Manufacturing” (see chapter 2.4). A deeper description can be found in the Submodel specification.

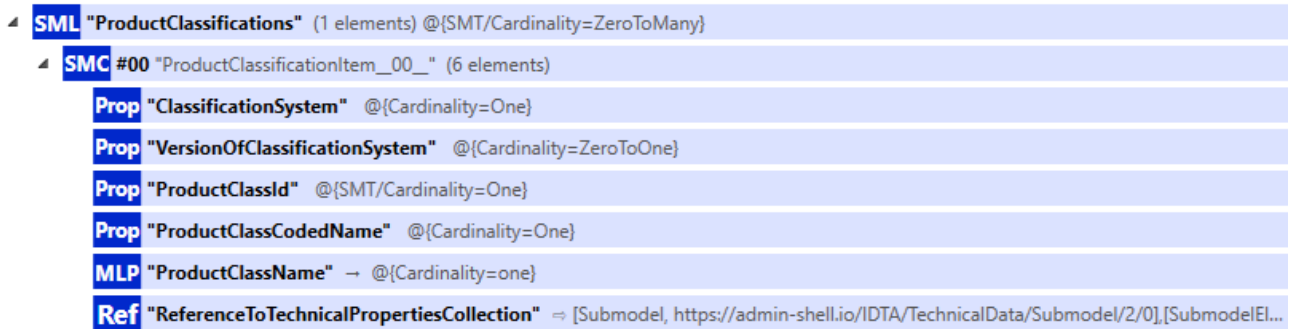


Figure 3: Submodel “Technical Data” with the SML “ProductClassifications” – Screenshot out of the Submodel description

Description of the elements

- [SML] SubmodelElementList “ProductClassifications”
  - Container of all product classifications that are structured in the next level as a SMC
  - multiple and unlimited product classifications are possible (cardinality)
- [SMC] SubmodelElementCollection “ProductClassificationItem”
  - group of properties to transport the classification and class information
  - one element of the cardinality
- [Prop] Property “ClassificationSystem”
  - Name of the classiciation system
  - mandatory
  - Fixed Naming: „ECLASS“
- [Prop] Property “VersionOfClassificationSystem”
  - Version of the classification system
  - optional (because company specific classifications often do not have a version information)
  - Fixed Naming: „14.0“ or „13.0“ or „10.0.1“
- [Prop] Property “ProductClassId”
  - IRDIs corresponding to the URI path concept (chapter 3.3)
    - IRDI of the ClassificationClass in value
    - IRDI of the ApplicationClass in valueId
  - mandatory
- [Prop] Property “ProductClassCodedName”
  - ClassCodedName (CCN) without hypen (-)
  - mandatory
- [MLP] MultiLanguageProperty “ProductClassName”
  - name of the class minimum in Englisch, multiple languages are possible
  - mandatory
- [Ref] ReferenceElement “ReferenceToTechnicalPropertiesCollection”
  - The Submodel “Technical Data” can transport multiple product classifications and also multiple technical property areas. This reference represents a connection between the elements from both areas.
  - mandatory if a classification in combination with a technical property area will be transported

The corresponding aasx file contains an example of a completely filled ECLASS class which is also shown in Figure 5.

5 Property Data Types

ECLASS and the AAS support different data types of properties. In addition it is possible / necessary to transport more than one value for a property (e.g. in context of a multivalue list). The different data types from the perspective of ECLASS are compared in the following table and also described in detail in this chapter.

ECLASS property data type	AAS property data type	Chapter
REAL_MEASURE	REAL_MEASURE	5.1
REAL_COUNT	REAL_COUNT	5.2
REAL_CURRENCY	REAL_CURRENCY	5.3
INTEGER_COUNT	INTEGER_COUNT	5.4
INTEGER_MEASURE	INTEGER_MEASURE	5.5
INTEGER_CURRENCY	INTEGER_CURRENCY	5.6
RATIONAL_COUNT	RATIONAL	5.7
RATIONAL_MEASURE	RATIONAL_MEASURE	5.8
BOOLEAN	BOOLEAN	5.9
STRING	STRING	5.10
STRING_TRANSLATABLE	STRING_TRANSLATABLE	5.11
DATE	DATE	5.12
TIME	TIME	5.13
TIMESTAMP	TIMESTAMP	5.14
URI	IRI	5.15
FILE	FILE	5.16
BLOB	BLOB	5.17

Table 2: ECLASS and AAS property data types

5.1 REAL_MEASURE

The <property> in <Submodel> transports mainly the value information and the corresponding <conceptDescription> transports the dictionary information in context of IEC 61 360.

5.1.1 REAL_MEASURE (single value)

Example element

Prop **"diameter"** = 20 [mm] @{datatype=REAL_MEASURE (single value)}

Submodel XML

```
[1]     <property>
[2]       <idShort>diameter</idShort>
          <displayName>
             <langStringNameType>
[3]             <language>en</language>
[3]             <text>diameter</text>
             </langStringNameType>
```

```
        </displayName>
        <semanticId>
           <type>ExternalReference</type>
           <keys>
              <key>
                 <type>GlobalReference</type>
[4]              <value>https://api.eclass-cdp.com/0173-1-02-AAC895-008</value>
              </key>
           </keys>
        </semanticId>
[5]     <supplementalSemanticIds>
           <reference>
              <type>ExternalReference</type>
                 <keys>
                    <key>
                       <type>GlobalReference</type>
[5]                    <value>0173-1#02-AAC895#008</value>
                    </key>
                 </keys>
           </reference>
        </supplementalSemanticIds>
[6]     <valueType>xs:float</valueType>
[7]     <value>20</value>
      </property>
```

Submodel explanations

- [1] <property> is for this data type
- [2] <idShort> is stylized name of the property
  - it must be unique und follow the AAS specification
- [3] <displayName> is the preferredName of the property (multiple languages)
- [4] <sematicID> is the IRDI of the property (URI path notation)
- [5] <supplementalSemanticIds> is another way for the IRDI notation (see also chapter 3.3)
- [6] <valueType> is „xs:float“ or “xs:double”
- [7] <value> contains the numerous value

REAL_MEASURE properties can be also connected to a valuelist (for example 0173-1-02-AAN528-007). Please see chapter 5.11 (property type STRING) for more details.

ConceptDescription XML

```
      <conceptDescription>
[1]     <idShort>diameter</idShort>
[2]     <displayName>
        <langStringNameType>
          <language>en</language>
          <text>diameter</text>
        </langStringNameType>
        </displayName>
[3]     <id>https://api.cdp-eclass.com/0173-1-02-AAC895-008</id>
        <embeddedDataSpecifications>
        <embeddedDataSpecification>
          <dataSpecification>
           <type>ExternalReference</type>
           <keys>
             <key>
              <type>GlobalReference</type>
              <value>http://admin-shell.io/DataSpecificationTemplates/DataSpecification
                     IEC61360/3/0</value>
             </key>
           </keys>
          </dataSpecification>
```

```
          <dataSpecificationContent>
           <dataSpecificationIec61360>
[4]          <preferredName>
              <langStringPreferredNameTypeIec61360>
                <language>en</language>
                <text>diameter</text>
              </langStringPreferredNameTypeIec61360>
              <langStringPreferredNameTypeIec61360>
                <language>de</language>
                <text>Durchmesser</text>
              </langStringPreferredNameTypeIec61360>
             </preferredName>
[5]          <unit>mm</unit>
             <unitId>
              <type>ExternalReference</type>
              <keys>
                <key>
                 <type>GlobalReference</type>
[6]              <value>0173-1#05-AAA480#004</value>
                </key>
              </keys>
             </unitId>
[7]          <dataType>REAL_MEASURE</dataType>
[8]          <definition>
              <langStringDefinitionTypeIec61360>
                <language>en</language>
                <text>extension, measured as spacing...</text>
              </langStringDefinitionTypeIec61360>
              <langStringDefinitionTypeIec61360>
                <language>de</language>
                <text>Ausdehnung, gemessen als Abstand...</text>
              </langStringDefinitionTypeIec61360>
             </definition>
           </dataSpecificationIec61360>
          </dataSpecificationContent>
        </embeddedDataSpecification>
         </embeddedDataSpecifications>
       </conceptDescription>
```

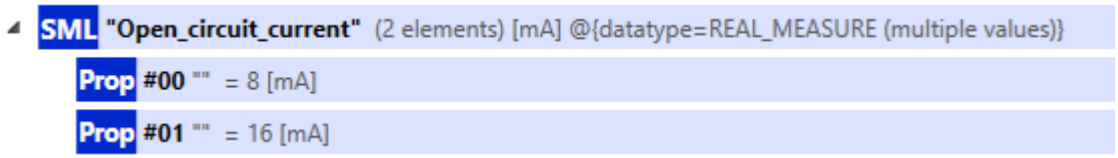
ConceptDescription explanations

- [1] <idShort> ist the identifyer of the conceptDescription
- [2] <displayName> is the preferredName of the property (multiple languages)
- [3] <id> is the IRDI of the property
    - the reference is set from the Submodel <semanticId><value> to the conceptDescription <id>
- [4] <proferredName> is the ECLASS preferredName of the property
- [5] <unit> is the ECLASS shortName of the unit
- [6] <unitId> ist the IRDI of the unit
- [7] <datatype> is REAL_MEASURE
- [8] <definition is the ECLASS definition of the property

5.1.2 REAL_MEASURE (multiple values)

Multiple values of the feature data type are combined in a SubmodelElementList SML element. The reason is that a <property> can only transport one value.

Example elements



Submodel XML

```
        <SubmodelElementList>
          <idShort>Open_circuit_current</idShort>
          <displayName>
             <langStringNameType>
                <language>en</language>
                <text>Open circuit current</text>
             </langStringNameType>
          </displayName>
[1]       <semanticId>
             <type>ExternalReference</type>
             <keys>
                <key>
                   <type>GlobalReference</type>
                   <value>https://api.cdp-eclass.com/0173-1-02-BAD858-006</value>
                </key>
             </keys>
          </semanticId>
[2]       <typeValueListElement>Property</typeValueListElement>
[2]       <valueTypeListElement>xs:float</valueTypeListElement>
          <value>
             <property>
[3]             <idShort/>
                <displayName>
                   <langStringNameType>
                      <language>en</language>
                      <text>Open circuit current</text>
                   </langStringNameType>
                </displayName>
                <semanticId>
                   <type>ExternalReference</type>
                   <keys>
                      <key>
                         <type>GlobalReference</type>
[1]                      <value>https://api.cdp-eclass.com/0173-1-02-BAD858-006</value>
                      </key>
                   </keys>
                </semanticId>
                <valueType>xs:float</valueType>
                <value>8</value>
             </property>
             <property>
                ...
             </property>
          </value>
        </SubmodelElementList>
```

Submodel explanations

- [1] <semanticID> is the same for the SML and the Properties
- [2] <typeValueListElement> and <valueTypeListElement> define the type of the included elements
  - it is only possible to use the same types in a SML
- [3] <idShort> is empty

ConceptDescription

The ConceptDescription only contains information about the property and not about the value. So the ConceptDescription can be used for for the SML and the Property as well.

5.2 REAL_COUNT

The property data type “REAL_COUNT” is used in the same way as “REAL_MEASURE” described in chapter 5.1 with the following differences.

ConceptDescription changes

- [7] <datatype> = REAL_COUNT
- [5] / [6] REAL_COUNT properties normally do not have a unit

5.3 REAL_CURRENCY

The property data type “REAL_CURRENCY” is used in the same way as “REAL_MEASURE” described in chapter 5.1 with the following differences. The currency is not explicit defined in ECLASS.

ConceptDescription changes

- [7] <datatype> = REAL_CURRENCY

5.4 INTEGER_COUNT

The property data type “INTEGER_COUNT” is used in the same way as “REAL_MEASURE” described in chapter 5.1 with the following differences.

Submodel changes

- [5] <valueType> is “xs:integer”

ConceptDescription changes

- [7] <datatype> = INTEGER_COUNT
- [5] / [6] INTEGER_COUNT properties normally do not have a unit

5.5 INTEGER_MEASURE

The property data type “INTEGER_MEASURE” is used in the same way as “INTEGER_COUNT” described in chapter 5.4 with the following differences.

Example elements

ConceptDescription changes

- [7] <datatype> = INTEGER_MEASURE

5.6 INTEGER_CURRENCY

The property data type “INTEGER_CURRENCY” is used in the same way as “INTEGER_COUNT” described in chapter 5.4 with the following differences.

ConceptDescription changes

- [7] <datatype> = INTEGER_CURRENCY

5.7 RATIONAL_COUNT

The property data type “RATIONAL_COUNT” is used in the same way as “REAL_MEASURE” described in chapter 5.1 with the following differences.

Submodel changes

- [5] <valueType> is “xs:string”

ConceptDescription changes

- [7] <datatype> = RATIONAL
- [5] / [6] INTEGER_COUNT properties normally do not have a unit

5.8 RATIONAL_MEASURE

The property data type “RATIONAL_MEASURE” is used in the same way as “RATIONAL_COUNT” described in chapter 5.7 with the following differences.

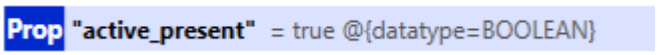
ConceptDescription changes

- [7] <datatype> = RATIONAL_MEASURE

5.9 BOOLEAN

ECLASS uses a restricted value list for the property data type BOOLEAN.

Example element



Submodel XML

```
       <property>
         <idShort>active_present</idShort>
[1]      <displayName>
            <langStringNameType>
               <language>en</language>
               <text>active present</text>
            </langStringNameType>
         </displayName>
[2]      <semanticId>
            <type>ExternalReference</type>
            <keys>
               <key>
                  <type>GlobalReference</type>
                  <value>https://api.cdp-eclass.com/0173-1-02-AAL309-007</value>
               </key>
```

```
            </keys>
         </semanticId>
[3]      <valueType>xs:boolean</valueType>
[4]      <value>true</value>
[5]      <valueId>
            <type>ModelReference</type>
            <keys>
               <key>
                  <type>ConceptDescription</type>
                  <value>https://api.cdp-eclass.com/0173-1-07-CAA016-001</value>
               </key>
            </keys>
         </valueId>
      </property>
```

Submodel explanations

- [1] <displayName> is the ECLASS preferredName
- [2] <semanticID> … <value> is the IRDI of the property
- [3] <valueType> is “xs:Boolean”
- [4] <value> can be “true” or “false”
- [5] <valueId> is the IRDI of the value

ConceptDescription XML

```
      <conceptDescription>
         <idShort>active_present</idShort>
[1]      <displayName>
            <langStringNameType>
               <language>en</language>
               <text>BOOLEAN</text>
            </langStringNameType>
         </displayName>
[2]      <id>https://api.cdp-eclass.com/0173-1-02-AAL309-007</id>
         <embeddedDataSpecifications>
            <embeddedDataSpecification>
               <dataSpecification>
                  <type>ExternalReference</type>
                  <keys>
                     <key>
                        <type>GlobalReference</type>
                        <value>http://admin-shell.io/DataSpecificationTemplates/DataSpecification
                               IEC61360/3/0</value>
                     </key>
                  </keys>
               </dataSpecification>
               <dataSpecificationContent>
                  <dataSpecificationIec61360>
[3]                  <preferredName>
                        <langStringPreferredNameTypeIec61360>
                           <language>en</language>
                           <text>active present</text>
                        </langStringPreferredNameTypeIec61360>
                        <langStringPreferredNameTypeIec61360>
                           <language>de</language>
                           <text>aktiv vorhanden</text>
                        </langStringPreferredNameTypeIec61360>
                     </preferredName>
[4]                  <unit/>
[5]                  <dataType>BOOLEAN</dataType>
[6]                  <definition>
                        <langStringDefinitionTypeIec61360>
                           <language>en</language>
                           <text>whether an ISDN distributor is active or not</text>
                        </langStringDefinitionTypeIec61360>
                        <langStringDefinitionTypeIec61360>
                           <language>de</language>
```

```
                        <text>Angabe, ob ein ISDN-Verteiler aktiv ist oder nicht</text>
                     </langStringDefinitionTypeIec61360>
                  </definition>
               </dataSpecificationIec61360>
            </dataSpecificationContent>
         </embeddedDataSpecification>
      </embeddedDataSpecifications>
   </conceptDescription>
```

ConceptDescription explanation

- [1] <displayName> is the ECLASS preferredName
- [2] <id> is the IRDI of the property
- [3] <preferredName> is the ECLASS preferredName
- [4] <unit> is empty
- [5] dataType> is “BOOLEAN”
- [6] <definition> is the ECLASS definition of the property

5.10 STRING

The property data type STRING is used to transport text values. STRING can be used in different ways:

- INDIRECT : valuelist for property which includes a defined list of values
- DIRECT : direct value für property, no IRDI available

Both ways can be used with one or multiple values. The cases are described in the following chapters.

5.10.1 STRING (1 value, indirect = value from a valuelist)

Example element

MLP **"pressure_measurement_variable_type"** → Relative pressure @{datatype=STRING (indirect, single value)}

Submodel XML

```
      <multiLanguageProperty>
        <idShort>pressure_measurement_variable_type</idShort>
[1]     <displayName>
           <langStringNameType>
              <language>en</language>
              <text>pressure measurement variable type</text>
           </langStringNameType>
        </displayName>
[2]     <semanticId>
           <type>ExternalReference</type>
           <keys>
              <key>
                 <type>GlobalReference</type>
[2]              <value>https://api.cdp-eclass.com/0173-1-02-AAO313-007</value>
              </key>
           </keys>
        </semanticId>
[3]     <value>
           <langStringTextType>
              <language>en</language>
[3]           <text>Relative pressure</text>
           </langStringTextType>
           <langStringTextType>
              <language>de</language>
              <text>Relativdruck</text>
           </langStringTextType>
        </value>
```

```
[4]      <valueId>
            <type>ExternalReference</type>
            <keys>
               <key>
                  <type>GlobalReference</type>
[4]               <value>https://api.cdp-eclass.com/0173-1-07-AAT462-001</value>
               </key>
            </keys>
         </valueId>
      </multiLanguageProperty>
```

Submodel explanations

- [1] <displayName> is the ECLASS preferredName
- [2] <semanticID> is the IRDI of the property
- [3] <value> is the preferredName of the value
- [4] <valueId> is the IRDI of the value out of the valuelist

ConceptDescription XML

```
      <conceptDescription>
         <idShort>STRING_INDIRECT__single</idShort>
[1]      <displayName>
            <langStringNameType>
               <language>en</language>
               <text>pressure measurement variable type</text>
            </langStringNameType>
            <langStringNameType>
               <language>de</language>
               <text>Messgrößenart (Druck)</text>
            </langStringNameType>
         </displayName>
[2]      <id>https://api.cdp-eclass.com/0173-1-02-AAO313-007</id>
         <embeddedDataSpecifications>
            <embeddedDataSpecification>
               <dataSpecification>
                  <type>ExternalReference</type>
                  <keys>
                     <key>
                        <type>GlobalReference</type>
                        <value>http://admin-shell.io/DataSpecificationTemplates/DataSpecification
                               IEC61360/3/0</value>
                     </key>
                  </keys>
               </dataSpecification>
               <dataSpecificationContent>
                  <dataSpecificationIec61360>
[3]                  <preferredName>
                        <langStringPreferredNameTypeIec61360>
                           <language>en</language>
                           <text>pressure measurement variable type</text>
                        </langStringPreferredNameTypeIec61360>
                        <langStringPreferredNameTypeIec61360>
                           <language>de</language>
                           <text>Messgrößenart (Druck)</text>
                        </langStringPreferredNameTypeIec61360>
                     </preferredName>
[4]                  <unit/>
[5]                  <dataType>STRING</dataType>
[6]                  <definition>
                        <langStringDefinitionTypeIec61360>
                           <language>en</language>
                           <text>type of the physical pressure variable to be measured</text>
                        </langStringDefinitionTypeIec61360>
                        <langStringDefinitionTypeIec61360>
                           <language>de</language>
```

```
                        <text>Art der physikalischen Druckgröße, die gemessen werden soll</text>
                     </langStringDefinitionTypeIec61360>
                  </definition>
               </dataSpecificationIec61360>
            </dataSpecificationContent>
         </embeddedDataSpecification>
      </embeddedDataSpecifications>
   </conceptDescription>
```

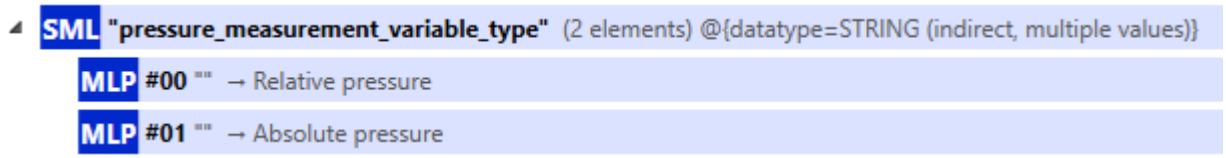
ConceptDescription explanation

- [1] <displayName> is the ECLASS preferredName
- [2] <id> is the IRDI of the property
- [3] <preferredName> is the ECLASS preferredName
- [4] <unit> is empty
- [5] dataType> is “STRING”
- [6] <definition> is the ECLASS definition of the property

5.10.2 STRING (n value, indirect = values from a valuelist)

As mentioned in chapter 5.1.2 Properties and MultiLanguageProperties cannot transport multiple values. So a SML is used as a container for multiple values here as well.

Example elements



5.10.3 STRING (1 value, direct = no valuelist, not transalatable)

This special form of the property data type is relevant for properties that do not have valuelists and are not translatable. An example is the “manufacturer part number”, which cannot be coosen from a valuelist and which also is the same in all languages. Example in ECLASS is IRDI 0173-1-02-AAO676-004.

Because the content is not multilanguage a <property> is used here, and not a <multiLanguageProperty>.

Example element

Submodel XML

```
      <property>
        <idShort>Internal_article_number</idShort>
        <semanticId>
           <type>ExternalReference</type>
           <keys>
              <key>
                 <type>GlobalReference</type>
                 <value>https://api.cdp-eclass.com/0173-1-02-AAO676-004</value>
              </key>
           </keys>
        </semanticId>
        <valueType>xs:string</valueType>
[1]     <value>A457123CC</value>
      </property>
```

Submodel explanation

- [1] <value> is the value that must be transported
- <valueId> is not used

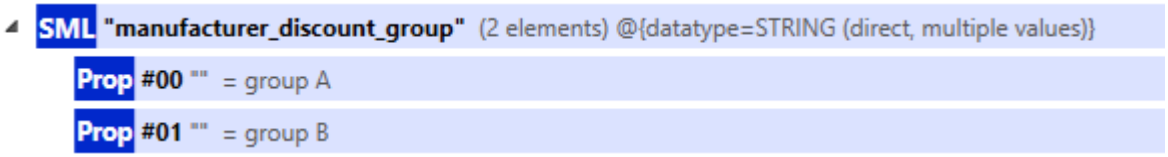
ConceptDescription

There are no changes in the ConceptDescription compared to chapter 5.10.1.

5.10.4 STRING (n values, direct = no valuelist, not transalatable)

This way follows the definition of the last three descriptions.

Example elements



5.11 STRING_TRANSLATABLE

The product data type STRING_TRANSLATABLE is used to transport direct values that can be delivered in different languages. This means that there isno IRDI available for the values.

5.11.1 STRING_TRANSLATABLE (single value)

Example elements

Submodel XML

```
[1]     <multiLanguageProperty>
          <idShort>Brand</idShort>
          <displayName>
             <langStringNameType>
                <language>en</language>
                <text>Brand</text>
             </langStringNameType>
          </displayName>
          <semanticId>
             <type>ExternalReference</type>
             <keys>
                <key>
                   <type>GlobalReference</type>
                   <value>https://api.cdp-eclass.com/0173-1-02-AAO742-002</value>
                </key>
             </keys>
          </semanticId>
          <value>
             <langStringTextType>
                <language>de</language>
                <text>Tesafilm</text>
             </langStringTextType>
             <langStringTextType>
                <language>en</language>
                <text>Tesa film</text>
             </langStringTextType>
          </value>
        </multiLanguageProperty>
```

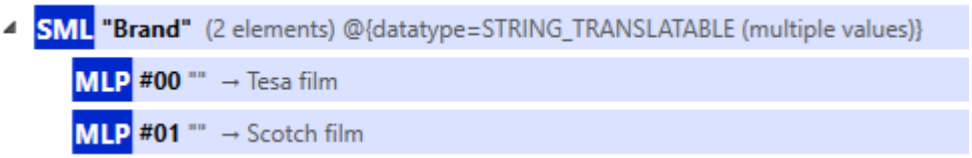
Submodel explanations

- [1] <multiLanguageProperty> is used because the values can be available in multiple languages
- <valueId> is not used

5.11.2 STRING_TRANSLATABLE (multiple values)

Here also a SML is used as a container for different values as described in chapter 5.1.2.

Example elements



5.12 DATE

The feature data type “DATE” is used in the same way as “STRING” (direct, single value) described in chapter 5.10.3 with the following differences in the Submodel.

- <valueType> = “xs:date”

In the ConceptDescription the following changes occur:

- <datatype> = “DATE”

Example element

5.13 TIME

The feature data type “TIME” is used in the same way as “STRING” (direct, single value) described in chapter 5.10.3 with the following differences in the Submodel

- <valueType> = “xs:time”

In the ConceptDescription the following changes occur:

- <datatype> = “TIME”

5.14 TIMESTAMP

The feature data type “TIMESTAMP” is used in the same way as “STRING” (direct, single value) described in chapter 5.10.3 with the following differences in the Submodel

- <valueType> = “xs:dateTime”

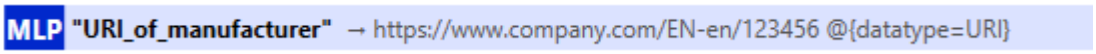
In the ConceptDescription the following changes occur:

- <datatype> = “TIMESTAMP”

5.15 URI

The feature data type URI allows the transport of URIs in the AAS and uses <MultiLanguageProperty> because URIs can be delivered in different languages.

Example element



Submodel XML

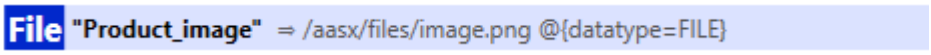
```
<multiLanguageProperty>
   <idShort>URI_of_manufacturer</idShort>
   <displayName>
      <langStringNameType>
         <language>en</language>
         <text>URI of manufacturer</text>
      </langStringNameType>
   </displayName>
   <semanticId>
      <type>ExternalReference</type>
      <keys>
         <key>
            <type>GlobalReference</type>
            <value>https://api.cdp-eclass.com/0173-1-02-ABA669-002</value>
         </key>
      </keys>
   </semanticId>
   <value>
      <langStringTextType>
         <language>de</language>
         <text>https://www.company.com/DE-de/123456</text>
      </langStringTextType>
      <langStringTextType>
         <language>en</language>
         <text>https://www.company.com/EN-en/123456</text>
      </langStringTextType>
   </value>
</multiLanguageProperty>
```

In the ConceptDescription the <datatype" is “STRING”.

5.16 FILE

The feature data type FILE allows the transport of files in the AAS.

Example element



Submodel XML

```
      <file>
        <idShort>Product_image</idShort>
        <displayName>
           <langStringNameType>
              <language>en</language>
              <text>Product image</text>
           </langStringNameType>
        </displayName>
        <semanticId>
           <type>ExternalReference</type>
           <keys>
              <key>
                 <type>GlobalReference</type>
                 <value>https://api.cdp-eclass.com/0173-1-02-ABK291-002</value>
```

```
              </key>
            </keys>
          </semanticId>
[1]       <value>/aasx/files/image.png</value>
[2]       <contentType>image/png</contentType>
        </file>
```

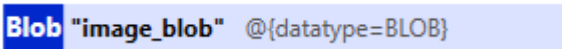
Submodel explanations

- [1] <value> refers the file in the aasx
- [2] <contentType> defines the contentype (list of defined values available)

5.17 BLOB

The feature data type BLOB allows the transport of BLOB information in the AAS.

Example element



Submodel XML

```
        <blob>
          <idShort>image_blob</idShort>
          <displayName>
            <langStringNameType>
              <language>en</language>
              <text>image as blob</text>
            </langStringNameType>
          </displayName>
          <semanticId>
            <type>ExternalReference</type>
            <keys>
              <key>
                <type>GlobalReference</type>
                <value>https://api.cdp-eclass.com/0173-1-02-XXX001-001</value>
              </key>
            </keys>
          </semanticId>
[1]       <value>YmxvYmRhdGFfYmxvYmRhdGE=</value>
[2]       <contentType>text/plain</contentType>
        </blob>
```

Submodel explanations

- [1] <value> contains the blob information as a Base64 coded bytestring
- [2] <contentType> is “text/plain”

6 ECLASS Advanced

Chapter 5 describes the ECLASS main elements (properties in different property data types) that are used in context of ECLASS Basic, Advanced and Asset. This chapter extends the description for ECLASS elements that are only used in ECLASS Advanced and Asset.

6.1 Aspect

ECLASS description

https://eclass.eu/support/technical-specification/structure-and-elements/aspect

ECLASS example

Aspects are used for structuring information in the ApplicationClass Advanced and also as a root node for Submodels in the ApplicationClass Asset.

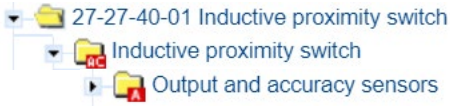


Figure 4: Aspect (in ECLASS CDP)

The details with IRDIs are shown below.

Structure	Description			
L L L L L L L L	Element	Element description	Value	IRDI
●	Class / Subgroup	27274001 / Inductive proximity switch	---	0173-1#01-AGZ376#021
●	ApplicationClass	Advanced	---	0173-1---ADVANCED_1_1-01-ADN934#013
●	Aspect	Output and accuracy sensors	---	0173-1#01-AHF601#002

Figure 5: Aspect (with details)

Element example

SMC "Output_and_accuracy_sensors" @{ECLASS=Aspect}

Submodel XML

```
[1]     <SubmodelElementCollection>
          <idShort>Output_and_accuracy_sensors</idShort>
          <displayName>
             <langStringNameType>
                <language>en</language>
                <text>Output and accuracy sensors</text>
             </langStringNameType>
          </displayName>
          <semanticId>
             <type>ExternalReference</type>
             <keys>
                <key>
                   <type>GlobalReference</type>
                   <value>https://api.cdp-eclass.com/0173-1-01-AHF601-002</value>
                </key>
             </keys>
          </semanticId>
        </submodelElementCollection>
```

Submodel explanation

- [1] <SubmodelElementCollection> is used for an ECLASS Aspect

ConceptDescription XML

```
        <conceptDescription>
         <idShort>Output_and_accuracy_sensors</idShort>
         <displayName>
            <langStringNameType>
               <language>en</language>
               <text>Output and accuracy sensors</text>
            </langStringNameType>
            <langStringNameType>
               <language>de</language>
               <text>Ausgang und Genauigkeit Sensorik</text>
            </langStringNameType>
         </displayName>
[1]      <id>https://api.cdp-eclass.com/0173-1-01-AHF601-002</id>
         <embeddedDataSpecifications>
            <embeddedDataSpecification>
               <dataSpecification>
                  <type>ExternalReference</type>
                  <keys>
                     <key>
                        <type>GlobalReference</type>
                        <value>https://admin-shell.io/DataSpecificationTemplates/DataSpecification
                               Iec61360/3/0</value>
                     </key>
                  </keys>
               </dataSpecification>
               <dataSpecificationContent>
                  <dataSpecificationIec61360>
[2]                  <preferredName>
                        <langStringPreferredNameTypeIec61360>
                           <language>en</language>
                           <text>Output and accuracy sensors</text>
                        </langStringPreferredNameTypeIec61360>
                        <langStringPreferredNameTypeIec61360>
                           <language>de</language>
                           <text>Ausgang und Genauigkeit Sensorik</text>
                        </langStringPreferredNameTypeIec61360>
                     </preferredName>
[3]                  <dataType>STRING</dataType>
[4]                  <definition>
                        <langStringDefinitionTypeIec61360>
                           <language>en</language>
                           <text>Output and accuracy sensors</text>
                        </langStringDefinitionTypeIec61360>
                        <langStringDefinitionTypeIec61360>
                           <language>de</language>
                           <text>Ausgang und Genauigkeit Sensorik</text>
                        </langStringDefinitionTypeIec61360>
                     </definition>
                  </dataSpecificationIec61360>
               </dataSpecificationContent>
            </embeddedDataSpecification>
         </embeddedDataSpecifications>
        </conceptDescription>
```

ConceptDescription Explanations

- [1] <id> contains the IRDI of the ECLASS Aspect
- [2] <preferredName> contains the ClassCodedName of the ECLASS Aspect
- [3] <datatype> is “STRING”
- [4] <definition> contains the definition of the ECLASS Aspect

6.2 Block

ECLASS description

https://eclass.eu/support/technical-specification/structure-and-elements/block

ECLASS example

Blocks are used to structure information in the ApplicationClass Advanced as well as in ApplicationClass Asset. They are always referenced by a reference property and can be nested. Blocks contain one or more properties, which can by regular properties, reference properties to other blocks, cardinality properties, or polymorphism properties.

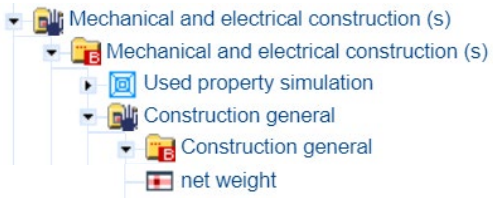


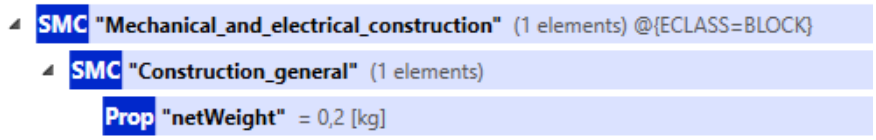
Figure 6: Block (in ECLASS CDP)

The details with IRDIs are shown below.

Structure	Description			
L L L L L L L L	Element name	Element description	Value	IRDI
	Class / Subgroup	27274001 / Inductive proximity switch	---	0173-1#01-AGZ376#021
	ApplicationClass	Advanced	---	0173-1---ADVANCED_1_1-01-ADN934#013
	RefProperty	Mechanical and electrical construction (s)	---	0173-1#02-AAR080#014
	Block	Mechanical and electrical construction (s)	---	0173-1#01-ADS444#014
	RefProperty	Construction general	---	0173-1#02-AAQ640#014
	Block	Construction general	---	0173-1#01-ADN455#014
	Property	Net weight	---	0173-1#02-AAF040#009

Figure 7: Block (with details)

Element example



Submodel XML

```
[1]     <SubmodelElementCollection>
          <idShort>Mechanical_and_electrical_construction</idShort>
          <displayName>
             <langStringNameType>
                <language>en</language>
                <text>Mechanical and electrical construction (s)</text>
             </langStringNameType>
          </displayName>
          <semanticId>
             <type>ExternalReference</type>
             <keys>
                <key>
                   <type>GlobalReference</type>
[2]                <value>https://api.cdp-eclass.com/0173-1-02-AAR080-014/0173-1-01-ADS444-
                         014</value>
                </key>
             </keys>
          </semanticId>
          <value>
[3]          <SubmodelElementCollection>
                <idShort>Construction_general</idShort>
```

```
                <displayName>
                   <langStringNameType>
                      <language>en</language>
                      <text>Construction general</text>
                   </langStringNameType>
                </displayName>
                <semanticId>
                   <type>ExternalReference</type>
                   <keys>
                      <key>
                         <type>GlobalReference</type>
[4]                      <value>https://api.cdp-eclass.com/0173-1-02-AAQ640-014/0173-1-01-ADN455-
                               014</value>
                      </key>
                   </keys>
                </semanticId>
                <value>
[5]                <property>
                      <idShort>netWeight</idShort>
                      <displayName>
                         <langStringNameType>
                            <language>en</language>
                            <text>net weight</text>
                         </langStringNameType>
                      </displayName>
                      <semanticId>
                         <type>ExternalReference</type>
                         <keys>
                            <key>
                               <type>GlobalReference</type>
                               <value>https://api.cdp-eclass.com/0173-1-02-AAF040-009</value>
                            </key>
                         </keys>
                      </semanticId>
                      <valueType>xs:float</valueType>
                      <value>0.2</value>
                   </property>
                </value>
             </SubmodelElementCollection>
          </value>
       </SubmodelElementCollection>
```

Submodel explanation

- [1] <SubmodelElementCollection> outer ReferenceProperty and Block
- [2] IRDIs of the ReferenceProperty and Block separated by / (see URI path in chapter 3.3)
- [3] <SubmodelElementCollection> of the integrated ReferenceProperty and Block
- [4] IRDIs of the ReferenceProperty and Block of the integrated elements
- [5] <property> in the Block

ConceptDescription

The ConceptDescription uses already described elements and is not described in detail.

6.3 Cardinality (in combination with a Block)

ECLASS description

https://eclass.eu/support/technical-specification/structure-and-elements/cardinality

ECLASS example

Cardinalities are used to multiply an ECLASS block in the output. So, a block (with always the same properties) can be used multiple times in the export. In ECLASS the number of the cardinality is defined in a separate property. This property is handled optional in the AAS and will not shown here.

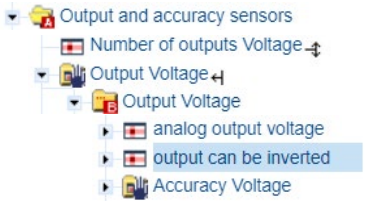


Figure 8: Cardinality with Block (in ECLASS CDP)

The details with IRDIs are shown below.

Structure	Description			
L L L L L L L L	Element	Element description	Value	IRDI
	Class / Subgroup	27274001 / Inductive proximity switch	---	0173-1#01-AGZ376#021
	ApplicationClass	Advanced	---	0173-1---ADVANCED_1_1-01-ADN934#013
	Aspect	Output and accuracy sensors	---	0173-1#01-AHF601#002
	RefProperty	Output voltage	---	0173-1#02-ABI555#003~x
	Block	Output voltage	---	0173-1#01-AHF842#003
	MultilangProp	analog output voltage	---	0173-1#02-ABI115#003
	Property	output can be inverted	---	0173-1#02-ABD898#003
	RefProperty	Accuracy Voltage	---	0173-1#02-ABI523#003
	Block	Accuracy Voltage	---	0173-1#01-AHF835#003

Figure 9: Cardinality with Block (with details)

Element example

Submodel XML

```
[1]     <S
ubmodelElementCollection>
         <idShort>Output_and_accuracy_sensors</idShort>
         <displayName>
            <langStringNameType>
               <language>en</language>
               <text>Output and accuray sensors</text>
            </langStringNameType>
         </displayName>
         <semanticId>
            <type>ExternalReference</type>
            <keys>
               <key>
                  <type>GlobalReference</type>
                  <value>https://api.cdp-eclass.com/0173-1-01-AHF601-002</value>
```

```
            </key>
         </keys>
      </semanticId>
      <value>
[2]      <SubmodelElementList>
            <idShort>Output_voltage</idShort>
            <displayName>
               <langStringNameType>
                  <language>en</language>
                  <text>Output voltage</text>
               </langStringNameType>
            </displayName>
            <semanticId>
               <type>ExternalReference</type>
               <keys>
                  <key>
                     <type>GlobalReference</type>
[3]                  <value>https://api.cdp-eclass.com/0173-1-02-ABI555-003/0173-1-01-AHF842-
                           003</value>
                  </key>
               </keys>
            </semanticId>
            <typeValueListElement>SubmodelElementCollection</typeValueListElement>
            <valueTypeListElement>xs:string</valueTypeListElement>
            <value>
[4]            <SubmodelElementCollection>
                  <idShort/>
                  <displayName>
                     <langStringNameType>
                        <language>en</language>
                        <text>Output voltage</text>
                     </langStringNameType>
                  </displayName>
                  <semanticId>
                     <type>ExternalReference</type>
                     <keys>
                        <key>
                           <type>GlobalReference</type>
[5]                        <value>https://api.cdp-eclass.com/0173-1-02-ABI555-003/0173-1-01-
                                 AHF842-003</value>
                        </key>
                        <key>
                           <type>GlobalReference</type>
[6]                        <value>https://api.cdp-eclass.com/0173-1-02-ABI555-003~0/0173-1-01-
                                 AHF842-003</value>
                        </key>
                     </keys>
                  </semanticId>
                  <value>
                     <multiLanguageProperty>
                        <idShort>analog_output_voltage</idShort>
                        <displayName>
                           <langStringNameType>
                              <language>en</language>
                              <text>analog output voltage</text>
                           </langStringNameType>
                        </displayName>
                        <semanticId>
                           <type>ExternalReference</type>
                           <keys>
                              <key>
                                 <type>GlobalReference</type>
                                 <value>https://api.cdp-eclass.com/0173-1-02-ABI115-003</value>
                              </key>
                           </keys>
                        </semanticId>
                        <value>
                           <langStringTextType>
                              <language>en</language>
                              <text>-10 ... +10 V</text>
```

```
                              </langStringTextType>
                              <langStringTextType>
                                 <language>de</language>
                                 <text>-10 ... +10 V</text>
                              </langStringTextType>
                           </value>
                           <valueId>
                              <type>ExternalReference</type>
                              <keys>
                                 <key>
                                    <type>GlobalReference</type>
                                    <value>https://api.cdp-eclass.com/0173-1-07-AAL491-004</value>
                                 </key>
                              </keys>
                           </valueId>
                        </multiLanguageProperty>
                        <property>
                           <idShort>output_can_be_inverted</idShort>
                           ...
                        </property>
                        <SubmodelElementCollection>
                           <idShort>Accuracy_Voltage</idShort>
                           ...
                        </SubmodelElementCollection>
                     </value>
                  </SubmodelElementCollection>
[7]               <SubmodelElementCollection>
                     <idShort/>
                     <semanticId>
                        <type>ExternalReference</type>
                        <keys>
                           <key>
                              <type>GlobalReference</type>
[8]                           <value>https://api.cdp-eclass.com/0173-1-02-ABI555-003~1/0173-1-01-
                                    AHF842-003</value>
                           </key>
                        </keys>
                     </semanticId>
                     <value>
                     ...
                     </value>
                  </SubmodelElementCollection>
[9]               <SubmodelElementCollection>
                     <idShort/>
                     <semanticId>
                        <type>ExternalReference</type>
                        <keys>
                           <key>
                              <type>GlobalReference</type>
[10]                          <value>https://api.cdp-eclass.com/0173-1-02-ABI555-003~2/0173-1-01-
                                    AHF842-003</value>
                           </key>
                        </keys>
                     </semanticId>
                     <value>
                     ...
                     </value>
                  </SubmodelElementCollection>
               </value>
            </SubmodelElementList>
         </value>
      </SubmodelElementCollection>
```

Submodel explanation

- [1] <SubmodelElementCollection> of the ECLASS Aspect
- [2] <SubmodelElementList> (SML) for the cardinality of multiple ECLASS Blocks

- [3] <value> IRDI of ReferenceProperty and Block in general
- [4] <SubmodelElementCollection> first cardinality Block in the SML
- [5] <value> of the first key of <sematicId> contains the IRDI combination of the ReferenceProperty and the Block separated by a / (see chapter 3.3)
- [6] <value> of the second key of <sematicId> contains the cardinality information with the numourus element ~0 after the ReferenceProperty
- [7] <SubmodelElementCollection> second cardinality Block in the SML
- [8] <value> IRDI of ReferenceProperty and Block of the second cardinaity element corresponding to URI path with the numourus element ~1
- [9] <SubmodelElementCollection> third cardinality Block in the SML
- [10] <value> IRDI of ReferenceProperty and Block of the third cardinaity element corresponding to URI path with the numourus element ~2

ConceptDescription

The ConceptDescription uses already described elements and is not described in detail.

6.4 Polymorphism (without Cardinality)

ECLASS description

https://eclass.eu/support/content-creation/content-development-platform/polymorphism-help-page

ECLASS example

This example shows an ECLASS polymorphism without cardinaity. In this case only one of the possible pathes can be chosen.

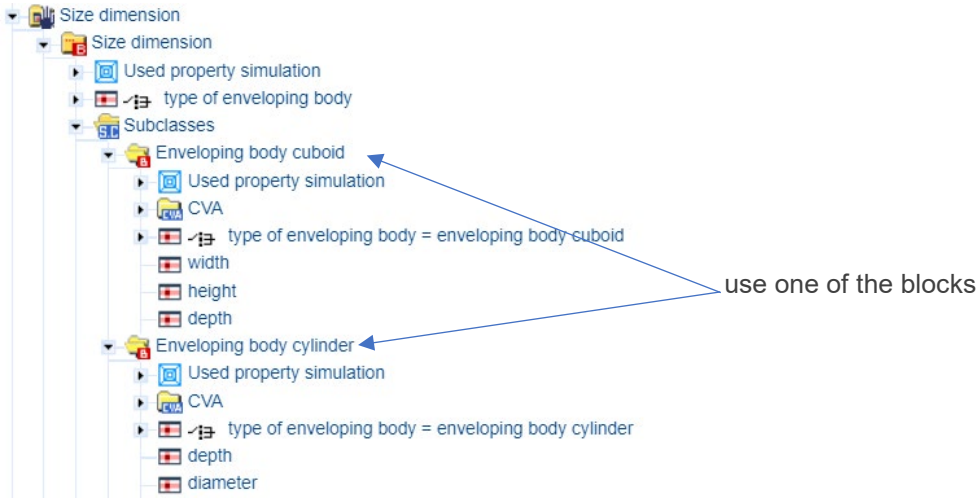
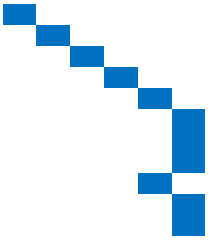


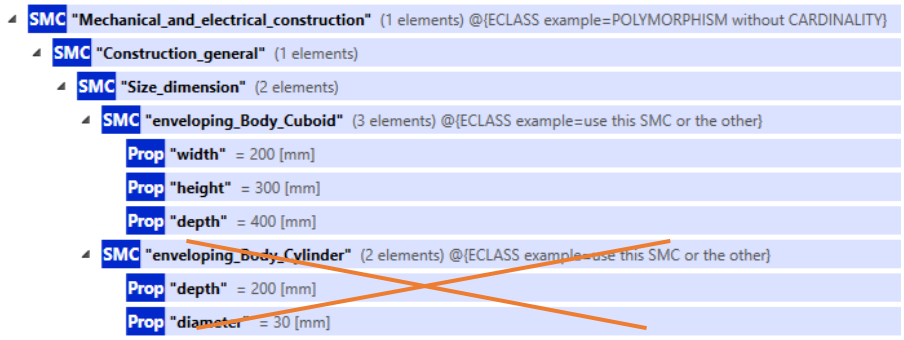
Figure 10: Polymorphism without cardinality (in ECLASS CDP)



Structure								Description			
L	L	L	L	L	L	L	L	Element name	Element description	Value	IRDI
								RefProperty	Construction general	---	0173-1#02-AAQ640#014
								Block	Construction general	---	0173-1#01-ADN455#014
								RefProperty	Size dimension	---	0173-1#02-AAQ667#006
								Block	Size dimension	---	0173-1#01-ADN458#006
								PolymorphBlock	Enveloping body cuboid	enveloping body cuboid	0173-1#01-ADS435#007
								Property	width	---	0173-1#02-BAF016#006
								Property	heigh	---	0173-1#02-BAA020#010
								Property	depth	---	0173-1#02-BAB577#008
								PolymorphBlock	Enveloping body cylinder	enveloping body cylinder	0173-1#01-ADS436#007
								Property	depth	---	0173-1#02-BAB577#008
								Property	diameter	---	0173-1#02-AAC895#007

Figure 11: Polymorphism without cardinality (with details)

Element example



Submodel

The Submodel only uses known elements so the XML code and descriptions are not shown here.

ConceptDescription

The ConceptDescription uses already described elements and is not described in detail.

6.5 Polymorphismus with Cardinality

ECLASS description

The same reference as in chapter 6.4 is used here.

https://eclass.eu/support/content-creation/content-development-platform/polymorphism-help-page

ECLASS example

This example shows an ECLASS polymorphism with cardinaity. Here multiple “physical quantities” can be choosen. All Blocks (under “subclasses”) are described by different properties.

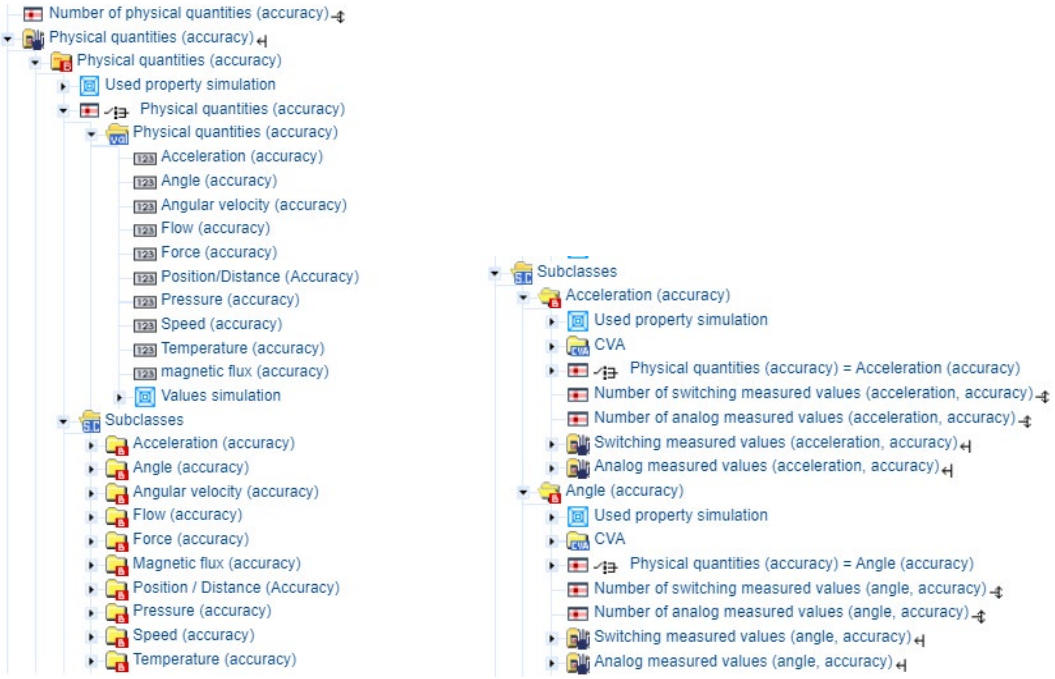


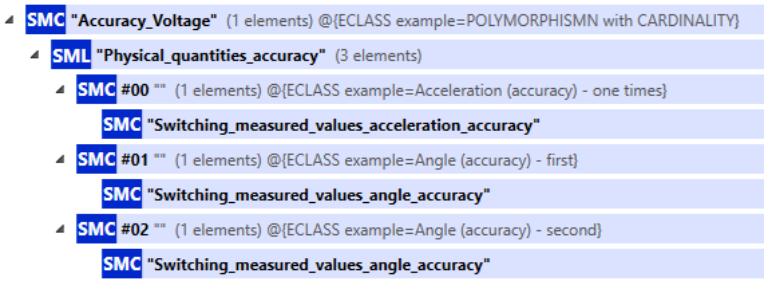
Figure 12: Polymorphism with cardinality (in ECLASS CDP)

Structure L L L L L L L L	Description Element name	Element description	Value	IRDI
	RefProperty	Accuray Voltage	---	0173-1#02-ABI523#003
	Block	Accuray Voltage	---	0173-1#01-AHF835#003
	RefProperty	Physical quantities (accuracy)	---	0173-1#02-ABI562#003
	Block	Physical quantities (accuracy)	---	0173-1#01-AHF845#003
	RefProperty	Acceleration (accuracy)	Acceleration (accuracy)	0173-1#02-ABI562#003
	Block	Acceleration (accuracy)	---	0173-1#01-AHW473#003
	RefProperty	Switching measured values (acceleration, accuracy)	---	0173-1#02-ABI539#003
	Block	Switching measured values (acceleration, accuracy)	---	0173-1#01-AHF804#003
	RefProperty	Angle (accuracy)	Angle (accuracy)	0173-1#02-ABI562#003~0
	Block	Angle (accuracy)	---	0173-1#01-AHW469#003
	RefProperty	Switching measured values (angle, accuracy)	---	0173-1#02-ABI531#003
	Block	Switching measured values (angle, accuracy)	---	0173-1#01-AHF796#003
	RefProperty	Angle (accuracy)	Angle (accuracy)	0173-1#02-ABI562#003~1
	Block	Angle (accuracy)	---	0173-1#01-AHW469#003
	RefProperty	Switching measured values (angle, accuracy)	---	0173-1#02-ABI531#003
	Block	Switching measured values (angle, accuracy)	---	0173-1#01-AHF796#003

Figure 13: Polymorphism with cardinality (with details)

If the polymorphism uses the same cardinality block more than once, the URI path notation for multiple elements is used for the ReferenceProperty (marked in red in the upper figure).

Element example



Submodel

The Submodel only uses known elements so the XML code and descriptions are not shown here.

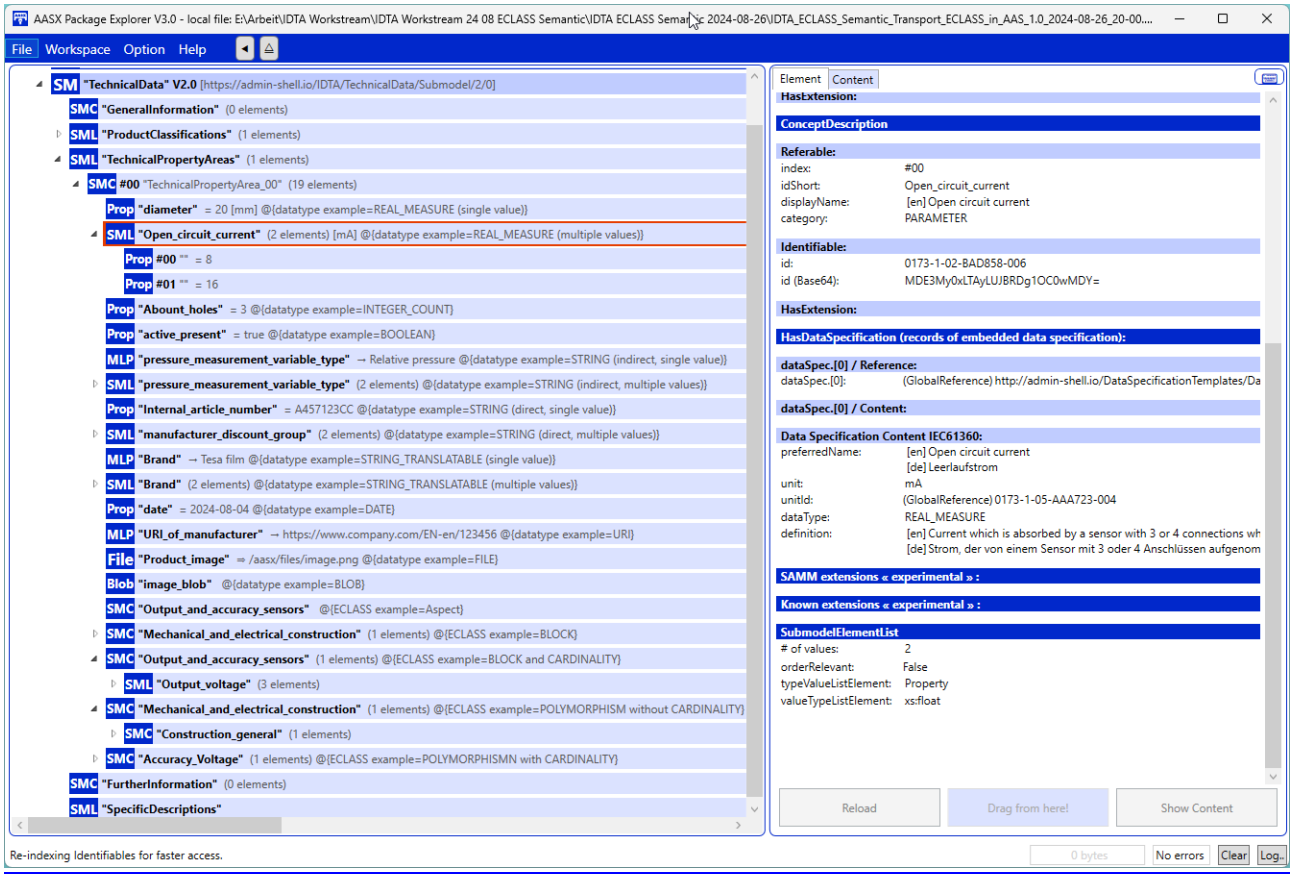
ConceptDescription

The ConceptDescription uses already described elements and is not described in detail.

7 Example Submodel

In combination with this document an aasx file is also delivered. The base of the example is the Submodel “Generic Frame for Technical Data for Industrial Equipment in Manufacturing” in version 2.0. So a practical approach is delivered.

To mark the property data types and also the ECLASS Advanced elements qualifies are used. Normally qualifiers are used in another context, but this useage allows to add visible comments in the Submodel.



www.industrialdigitaltwin.org
www.eclass.eu