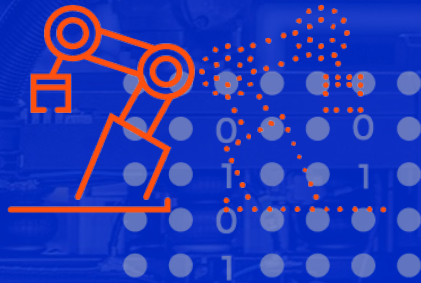
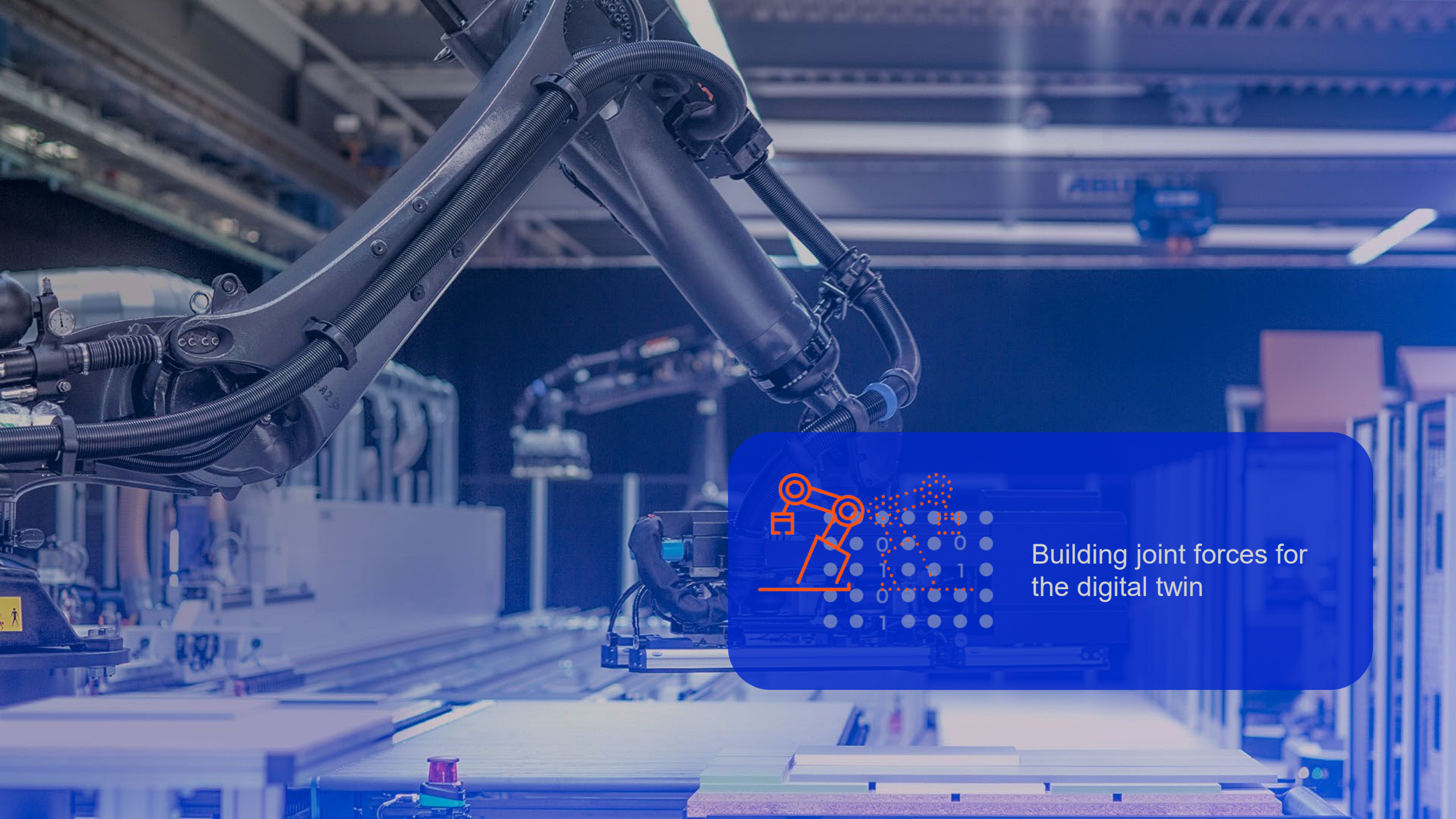


Tutorial

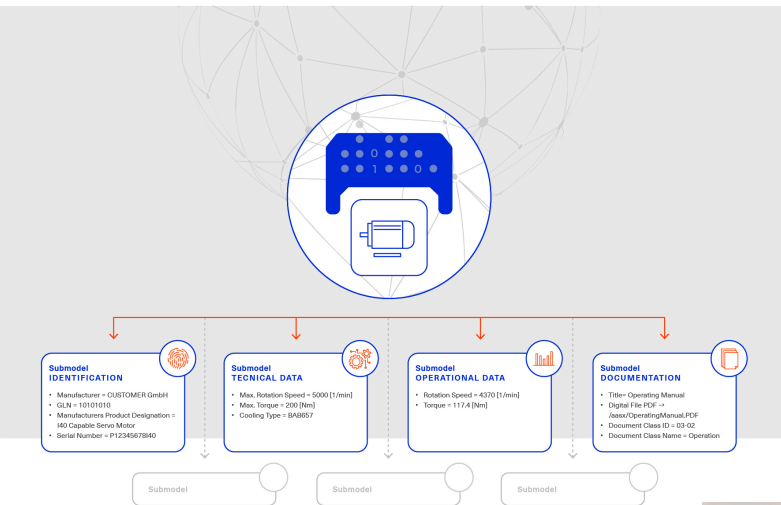
Specification Asset Administration Shell

Part 3a Data Specification IEC61360 v3.0 April 2023

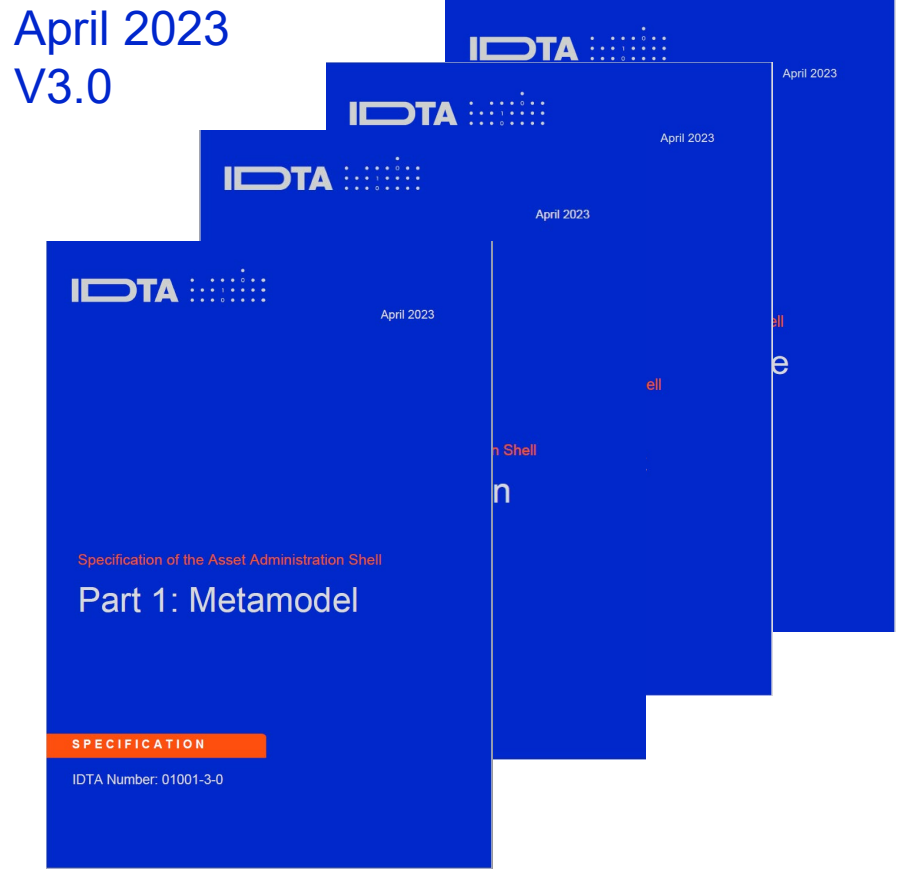


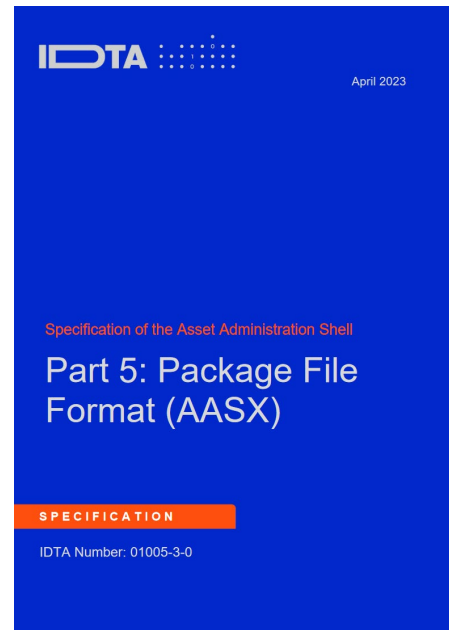
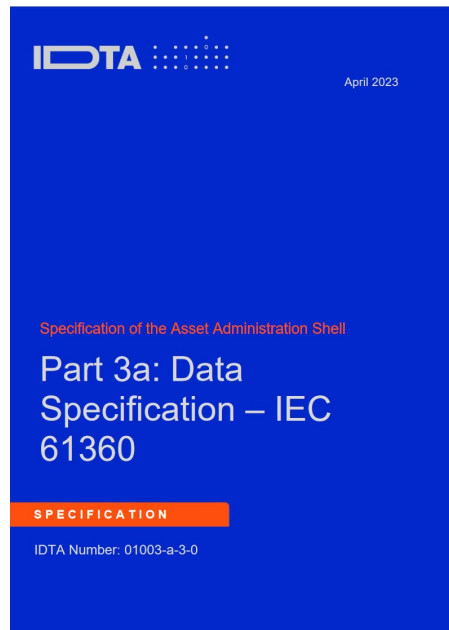
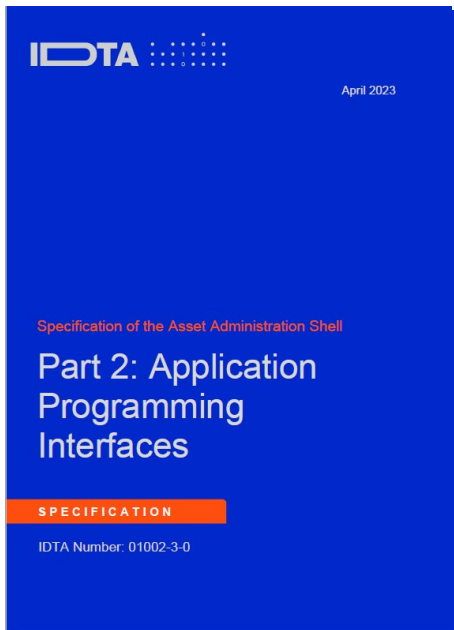
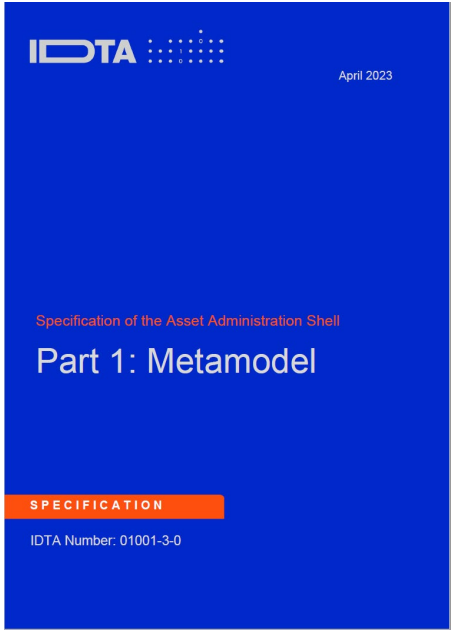
Building joint forces for
the digital twin

From Concept to Specification



2018
V1.0





V3.0
April 2023



IEC 63278



This Tutorial is
about Part 3a

For whom is this tutorial?



- You should have an idea of the benefits of introducing digital twins to your domain
- You should know Tutorial Part 1 Metamodel
- You should have basic knowledge in UML modeling
- You are an architect and want to learn more about the underlying information model of the Asset Administration Shell
- You are a developer and want to upgrade to the new version or start your first implementation

For whom is this tutorial?

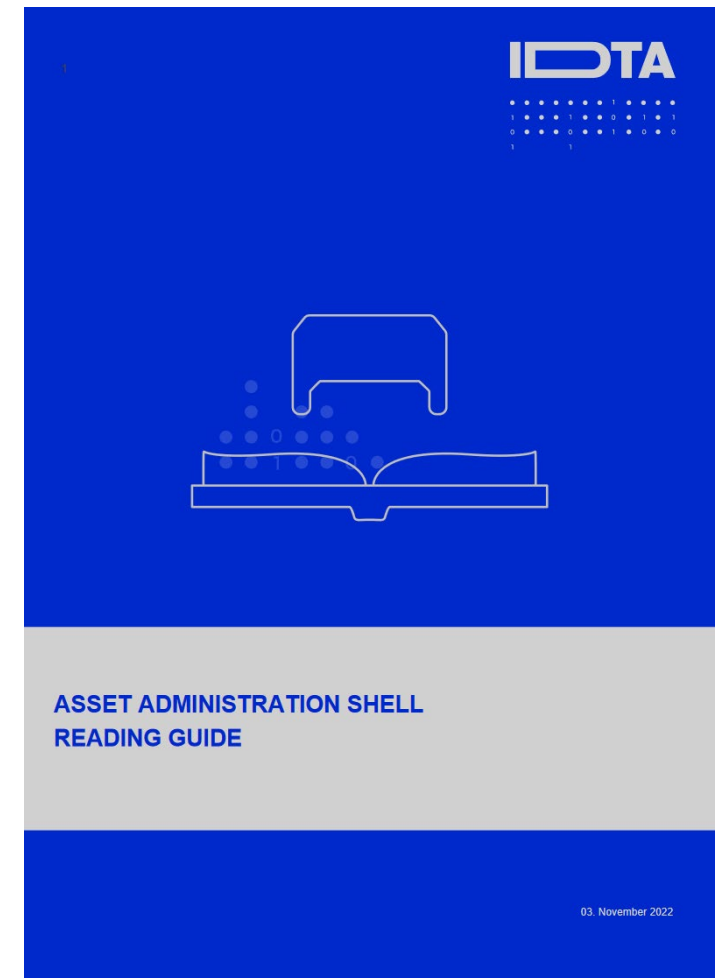


- You feel dissappointed and do not know how to start?

Have a look at the Asset Administration Shell Reading Guide!

It is updated on a regular basis.

- ▶ **Where to start:** If you have never heard of the AAS
- ▶ **For the generally interested reader:** If you want to learn more about the subject
- ▶ **For decision makers:** If you are interested in the business side of I4.0
- ▶ **For software developers and architects:** If you want to know how to create software for the AAS
- ▶ **For users of the AAS and domain experts:** If you are interested in using the AAS for specific tasks
- ▶ **Security and AI:** If you want to deep dive into these special topics.





● ● ● ● ● ● ● ●
● ● 0 ● ● 0 ● ●
● ● 1 ● ● 1 ● ●
● ● 0 ● ● ● ● ● ●
● ● 1 ● ● ● ● ● ●

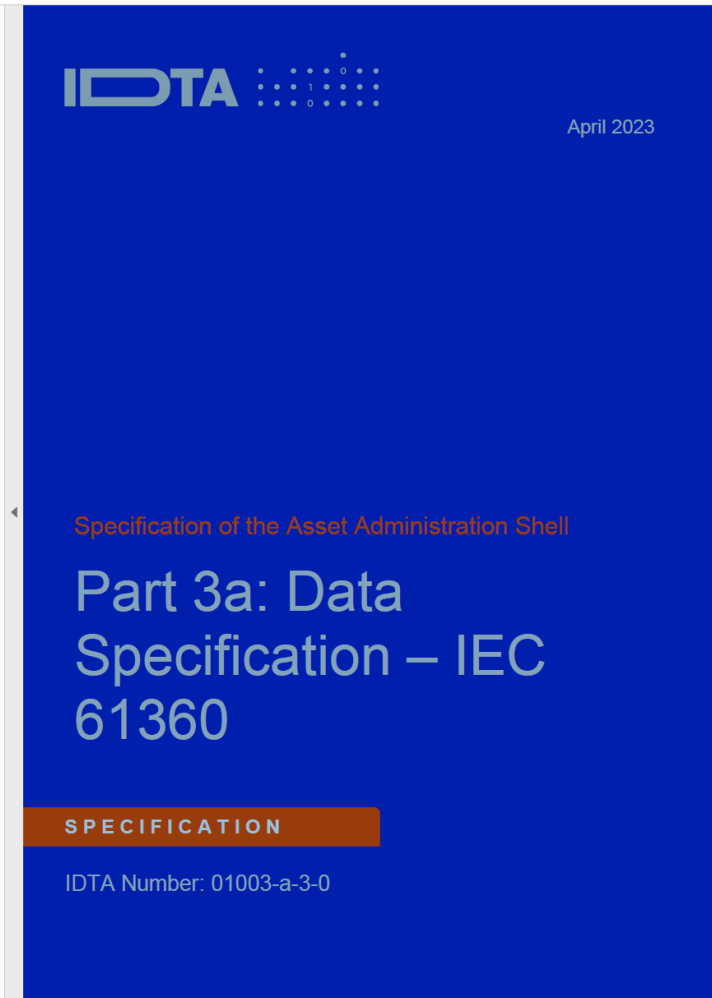
Get started

Download Specification

<https://industrialdigitaltwin.org/content-hub/>

Bookmarks

- > 1 Preamble
- > 2 Terms, Definitions and Abbreviations
- > 3 Introduction
- ▼ 4 General
 - 4.1 Introduction
 - 4.2 Concept Descriptions for Properties and Values
- ▼ 5 Predefined Data Specification Templates
 - 5.1 Overview
- ▼ 6 Predefined Template for IEC61360 Properties, Value Lists, and Values (normative)
 - 6.1 Data Specification IEC61360 Template Specification
 - 6.2 Data Specification IEC61360 Attributes
 - 6.3 Enumeration Data Type IEC61360
 - 6.4 Level Type
 - 6.5 Value List Attributes
- 7 Mapping IEC 61360 Data Types to XSD Data Types
- 8 Category of Concept Descriptions
- ▼ 9 Cross-Constraints and Invariants for Predefined Data Specifications (normative)
 - 9.1 General
 - 9.2 Constraints for Data Specification IEC61360
- ▼ 10 Primitive and Simple Data Types (normative)
 - 10.1 Predefined Simple Data Types
 - 10.2 Basic and Primitive Data Types
- > 11 Mappings to Data Formats to Share I4.0-Compliant Information (normative)
- 12 Summary and Outlook
- > Annex A. Background Information
- Annex B. Backus-Naur-Form
- > Annex C. Templates for UMI Tables



Import XMI to your UML tooling

1.



Releases 7

AAS Schemas V3.0.6 Latest
3 weeks ago

+ 6 releases

<https://github.com/admin-shell-io/aas-specs/releases>

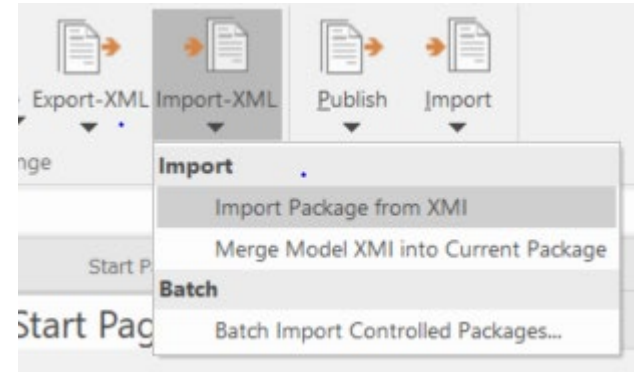
admin-shell-io / aas-specs

<> Code ! Issues 8 🔗 Pull requests

master aas-specs / schemas / xmi /

<https://github.com/admin-shell-io/aas-specs/tree/master/schemas/xmi>

2.



1. Fetch release of AAS you are interested in
2. Import xmi file into UML tool (best with Enterprise/Architect)

Note: The data specifications are embedded into the Part 1 schemas



- Annex E. Metamodel Changes
 - General
 - Changes V3.0 vs. Part 1 V2.0.1
 - Changes V3.0 vs. Part 1 V3.0RC02

Note for Experts: <Notes for tutorial listeners who have knowledge of previous versions of the specification (V2.0 or Release Candidates of V3.0).

If you do not know previous versions you can ignore these notes.>



14 - Birgit Boss: Details of the Asset Administration Shell V3.0 (Release...

V3.0RC02

<https://www.youtube.com/watch?v=QR-nOI6cuOI>

Annex D. Legend for UML Modelling

OMG UML General

This annex explains the UML elements used in this specification. For more information, please refer to the comprehensive literature available for UML. The formal specification can be found in [12].

Figure 20 shows a class with name "Class1" and an attribute with name "attr" of type *Class2*. Attributes are owned by the class. Some of these attributes may represent the end of binary associations, see also Figure 21. In this case, the instance of *Class2* is navigable via the instance of the owning class *Class1*.¹⁴

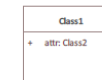


Figure 20 Class

Figure 21 shows that *Class4* inherits all member elements from *Class3*. Or in other words, *Class3* is a generalization of *Class4*, *Class4* is a specialization of *Class3*. This means that each instance of *Class4* is also an instance of *Class3*. An instance of *Class4* has the attributes *attr1* and *attr2*, whereas instances of *Class3* only have the attribute *attr1*.

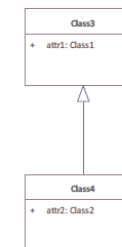


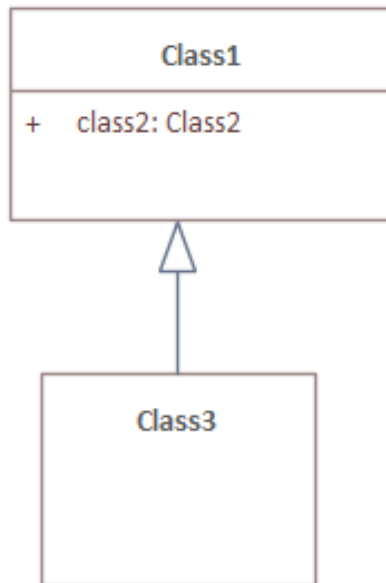
Figure 21 Inheritance/Generalization

Figure 22 defines the required and allowed multiplicity/cardinality within an association between instances of

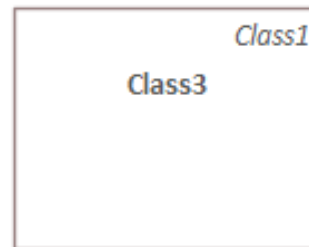
1. Get (re-)familiar with general UML modeling rules
2. Get familiar with specific graphical representation of UML in the specification (partly tool specific)



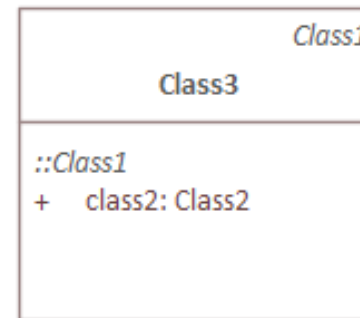
A)



B)



C)

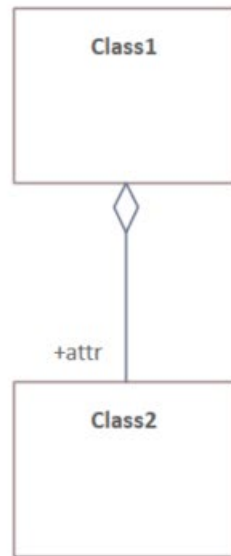


Hint: Graphical representation tool specific

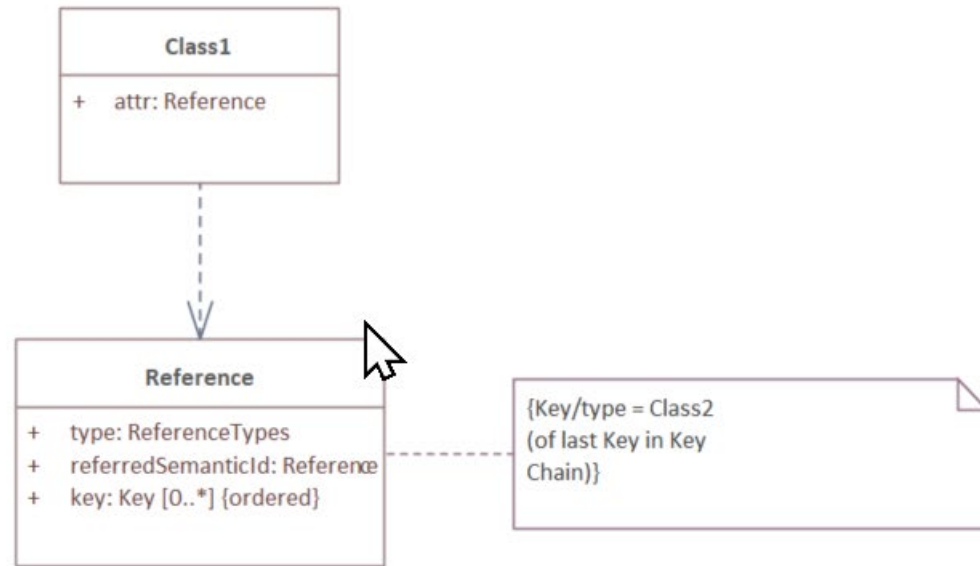


Note: Referencing of Referables is an important concept to understand when implementing the AAS

A)

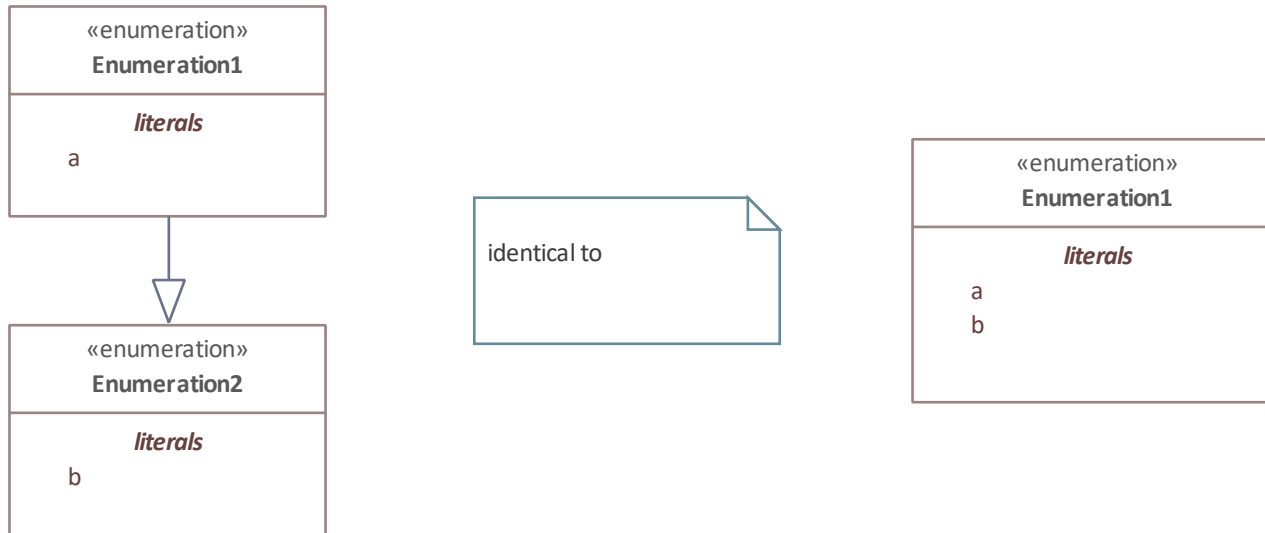


B)



Note for experts: In previous versions a notation of class attributes with reference (*) was used additionally to the notation with the association with the diamond.

Enumerations



Note 1: Inheritance between enumerations is not widely used. It is only used for graphical illustration of relationships between enumerations



Template for Classes

Template for Classes:

Class:	<Class Name> [<<abstract>>] ["<<Experimental>>"] ["<<Deprecated>>"] ["<<Template>>"]		
Explanation:	<Explanatory text>		
Inherits from:	{<Class Name> ";"}+ "-"		
Attribute	Explanation	Type	Card.
<attribute or association name> ["<<ordered>>"] ["<<Experimental>>"] ["<<Deprecated>>"]	<Explanatory text>	<Type>	<Card>

Note for experts:
ModelReference<SubmodelElement> is equal to former notation SubmodelElement*

Note for experts: no kind column any longer, instead different notation for Type

- v
🔖 Annex D. Templates for UML Tables
 - 🔖 General
 - 🔖 Template for Classes
 - 🔖 Template for Enumerations
 - 🔖 Template for Primitives
 - 🔖 Handling of Constraints

Example for Class Specification

5.3.7.12 Property Attributes

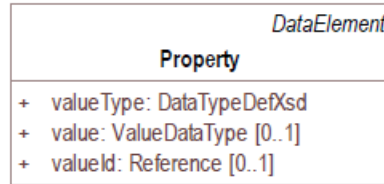


Figure 39 Metamodel of Properties

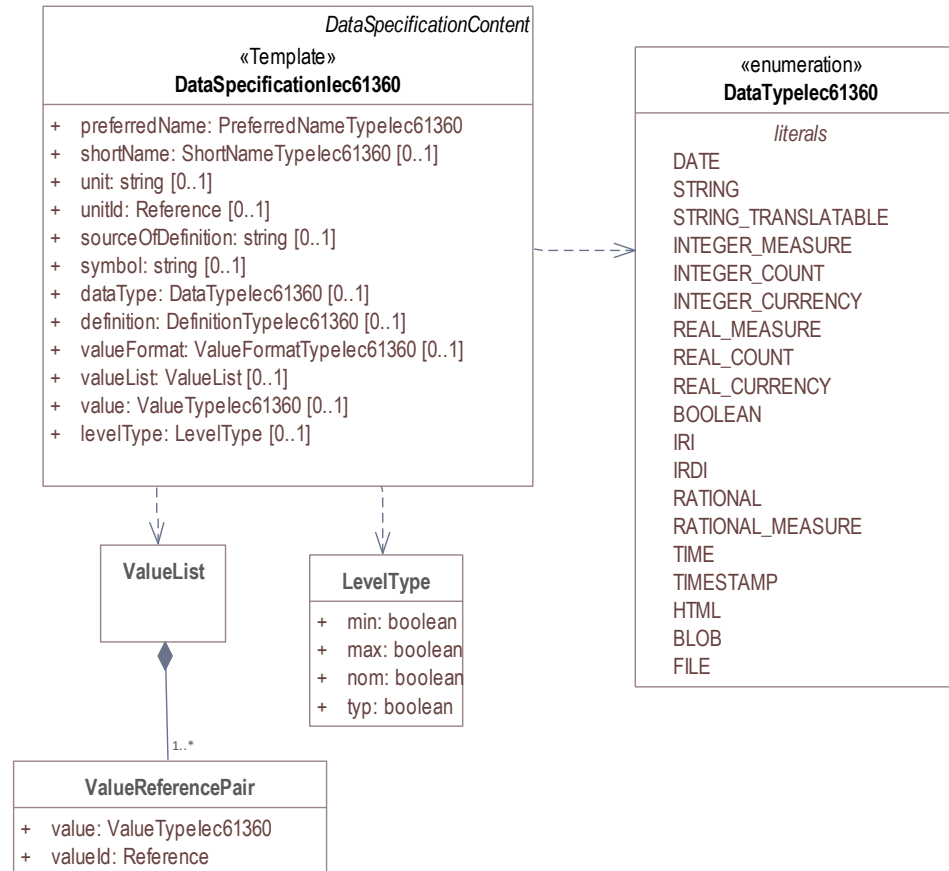
Class:	Property		
Explanation:	<p>A property is a data element that has a single value.</p> <p><u>Constraint AASd-007:</u> If both the <i>Property/value</i> and the <i>Property/valueId</i> are present, the value of <i>Property/value</i> needs to be identical to the value of the referenced coded value in <i>Property/valueId</i>.</p>		
Inherits from:	DataElement		
Attribute	Explanation	Type	Card.
valueType	Data type of the value attribute	DataTypeDefXsd	1
value	The value of the property instance	ValueDataType	0..1
valueId	Reference to the global unique ID of a coded value	Reference	0..1
	Note: it is recommended to use an external reference.		



●	●	●	●	●	●	●	●
●	0	●	0	●	●	●	●
●	1	●	1	●	●	●	●
●	0	●	●	●	●	●	●
●	1	●	●	●	●	●	●

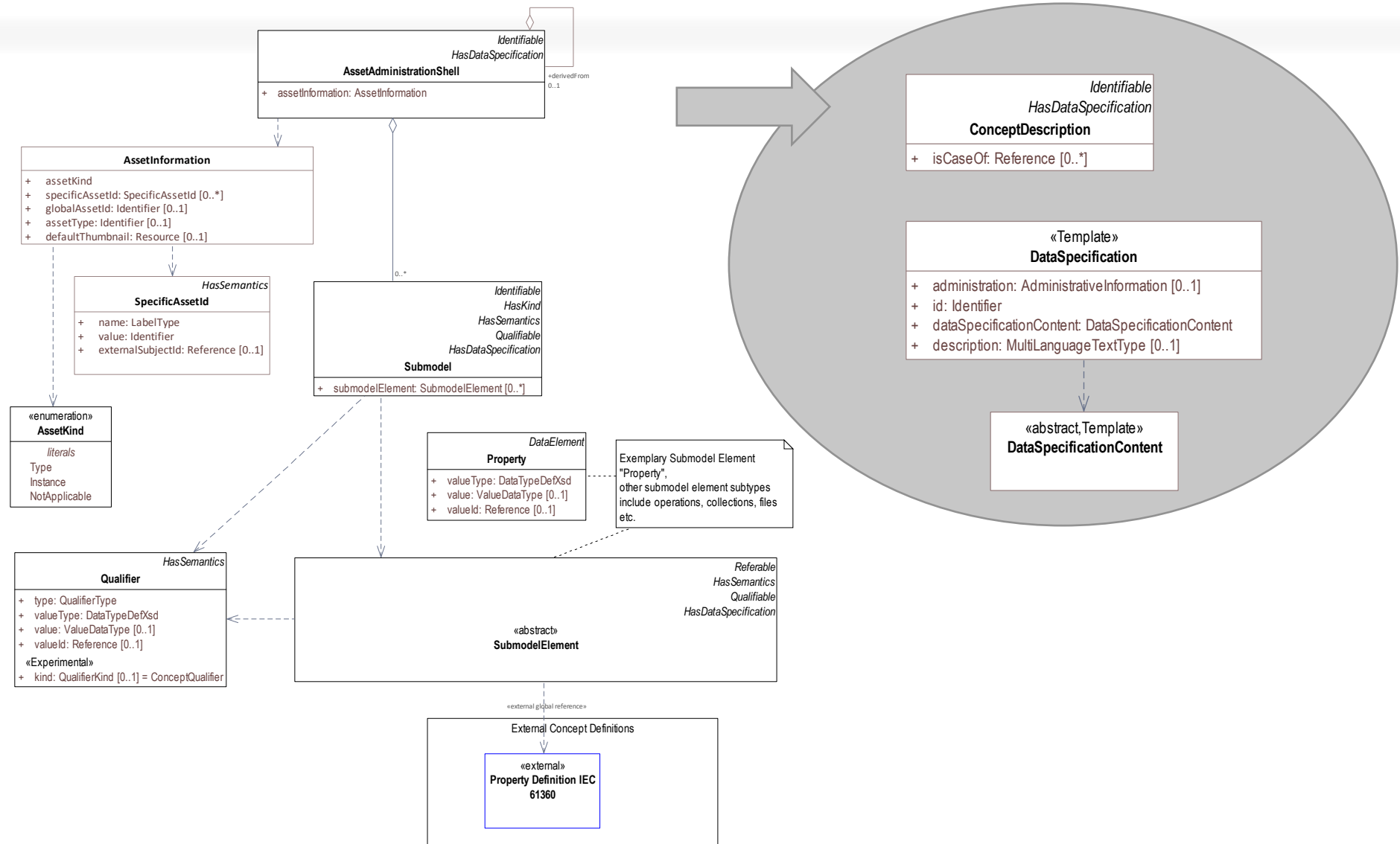
Get warm – Prerequisites
from PART 1 Metamodel

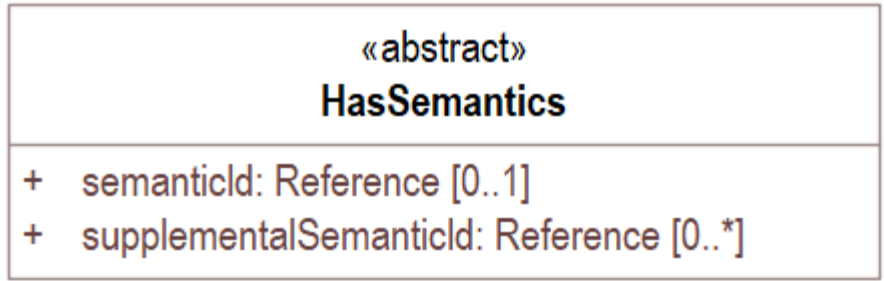
Overview – the goal is to be able to read this diagram



Data Specification
for Concept
Descriptions (see
Part 1)

Part 1 – Data Specifications & Concept Descriptions





The semanticId is the identifier of the semantic definition of the element.
 Supplemental semantic IDs can be added.
 The semanticId can be an External Reference or a Model Reference to a **Concept Description**.





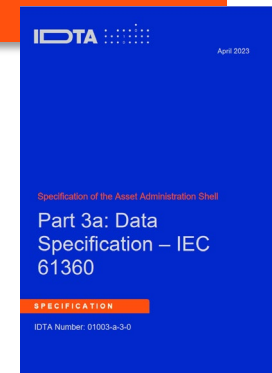
«abstract» HasDataSpecification
+ dataSpecification: Reference [0..*]

Allows to define standardized templates for data specification

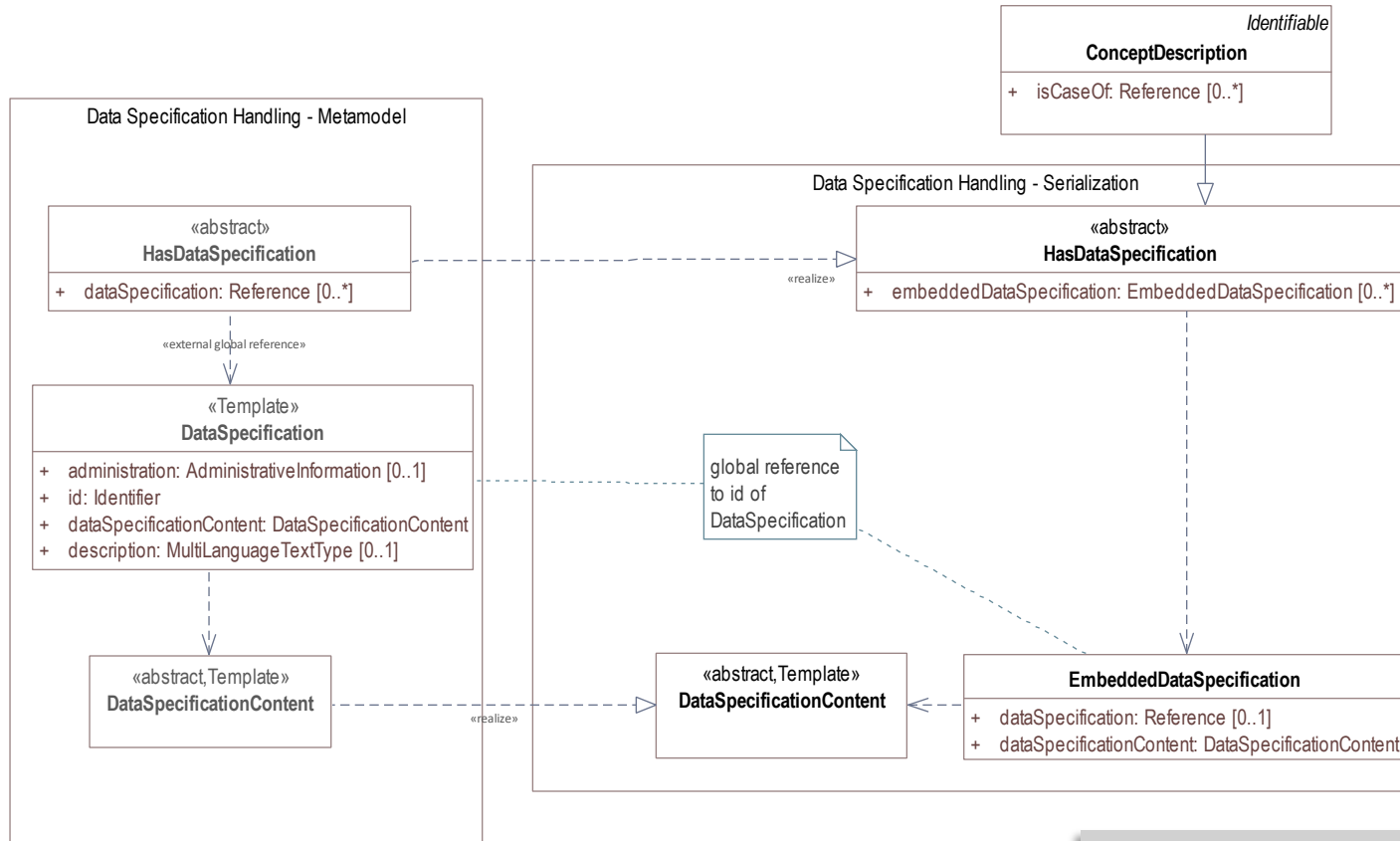
Data specification reference shall be globally unique and identifies which data specifications are used for an object

Attributes defined in template are added to the object

Note for Experts: Data Specifications are not part of Part 1 any longer: They are part of data specifications series Part 3



Part 1 - Embedded Data Specifications



Property	0173-1#02-BAA120#008 Max. rotation speed
Data type	INTEGER_MEASURE
Unit of measure	1/min
Definition	Greatest possible rotation speed with which the motor or feeding unit may be operated

<i>DataSpecificationContent</i>	
«Template»	
DataSpecificationIec61360	
+ preferredName:	PreferredNameTypeIec61360
+ shortName:	ShortNameTypeIec61360 [0..1]
+ unit:	string [0..1]
+ unitId:	Reference [0..1]
+ sourceOfDefinition:	string [0..1]
+ symbol:	string [0..1]
+ dataType:	DataTypeInfoIec61360 [0..1]
+ definition:	DefinitionTypeInfoIec61360 [0..1]
+ valueFormat:	ValueFormatTypeIec61360 [0..1]
+ valueList:	ValueList [0..1]
+ value:	ValueTypeIec61360 [0..1]
+ levelType:	LevelType [0..1]

In formats like xml, JSON, rdf the embedded data specification approach is implemented

Data Specifications in Part 3 – data specification template IEC 61360 defined in Part 3a

Administration shell

Serial number
Inverter current

i700E70

Administration shell

231231 Serial number
0.02 Inverter current

● ● ● ● ● ● ● ●
● ● 0 ● ● 0 ● ●
● ● 1 ● ● 1 ● ●
● ● 0 ● ● ● ● ● ●
● ● 1 ● ● ● ● ● ●

IEC 61360

IEC 61360

[Add languages](#) ▾

[Article](#) [Talk](#)

[Read](#) [Edit](#) [View history](#) [Tools](#) ▾

From Wikipedia, the free encyclopedia

IEC 61360, with the title "Standard data element types with associated classification scheme", is a series of [standard documents](#) defining a general purpose vocabulary in terms of a reference dictionary published by the [International Electrotechnical Commission](#).

Intended use [\[edit \]](#)

The vocabulary specified in IEC 61360 may be used to define [ontologies](#) for use in the field of [electrotechnology](#), electronics and related domains.^[1]

Structure [\[edit \]](#)

The IEC 61360 series is structured into different parts:

- IEC 61360-1 - Part 1: Definitions - Principles and methods
- IEC 61360-2 - Part 2: EXPRESS dictionary schema
- IEC 61360-4 - Part 4: IEC Common Data Dictionary (IEC CDD)
- IEC 61360-6 - Part 6: IEC Common Data Dictionary (IEC CDD) quality guidelines

IEC 61360-1 provides a detailed introduction to the structure of the dictionary and its use.^[1] IEC 61360-2 specifies the detailed dictionary data model and IEC 61360-6 stipulates quality criteria for the content of the dictionary.^[1] The data model defined in IEC 61360-2 is also published in ISO 13584-42.^[2]

The IEC provides a technical dictionary for the use in the electro-technical and electronic domain which is published as IEC 61360-4. This dictionary is called [IEC Common Data Dictionary \(IEC CDD\)](#) and can be accessed as a web page (<https://cdd.iec.ch>).

See also [\[edit \]](#)

IEC 61360 also defines the base for other product taxonomies like [eCl@ss](#).

[Industrie 4.0](#) uses product property description based on IEC 61360.^[3]

- ["IEC 61360"](#) at [International Electrotechnical Commission](#)

Note: Always be aware of usage policies and licensing before using a predefined/standardized concept description!

Status 2023-09



Preferred name	Max. rotation speed
IRDI	0173-1#02-BAA120#008
Definition	Greatest permissible rotation speed with which the motor or feeding unit may be operated
Short name of unit	1/min
Quantity	frequency
Type of Property	Non-dependent
Valency type	Multivalent
Definition class	ECLASS (0173-1#01-RAA001#001)
Property data type	Integer (measure)
Class type code	F03 - frequency, rotational frequency
Allow negative values	false
Property Original Identifier	BAA120001

Example
measurable
Property from
ECLASS



Property	02-BAE122 Cooling type
short name	-
Format	STRING
Definition:	Summary of various types of cooling, for use as search criteria that limit a selection
Values:	
	0173-1#07-BAB649#001 - Air-air heat exchanger
	0173-1#07-BAB650#001 - Air-water heat exchanger
	0173-1#07-BAB592#001 - alien
	0173-1#07-BAB611#001 - closed, external air-cooling
	0173-1#07-BAB610#001 - closed, internal air-cooling
	0173-1#07-BAB591#003 - free cooling
	0173-1#07-BAB702#003 - Heat exchanger against other cooling medium
	0173-1#07-BAB657#003 - open circuit, external cooling
	0173-1#07-BAB656#003 - open circuit, internal cooling
	0173-1#07-BAB535#003 - other form of cooling with primary air coolant
	0173-1#07-BAB536#003 - other primary non-air coolant
	0173-1#07-BAB674#003 - self

Example
Value List
from
ECLASS

IEC 61360 Value

Property	02-BAE122 Cooling type
short name	-
Format	STRING
Definition:	Summary of various types of cooling, for use as search criteria that limit a selection
Values:	
0173-1#07-BAB649#001 - Air-air heat exchanger	
0173-1#07-BAB650#001 - Air-water heat exchanger	
0173-1#07-BAB592#001 - alien	
0173-1#07-BAB611#001 - closed, external air-cooling	
0173-1#07-BAB610#001 - closed, internal air-cooling	
0173-1#07-BAB591#003 - free cooling	
0173-1#07-BAB702#003 - Heat exchanger against other cooling medium	
0173-1#07-BAB657#003 - open circuit, external cooling	
0173-1#07-BAB656#003 - open circuit, internal cooling	
0173-1#07-BAB535#003 - other form of cooling with primary air coolant	
0173-1#07-BAB536#003 - other primary non-air coolant	
0173-1#07-BAB674#003 - self	

Value	0173-1#07-BAB657#003
Classification	open circuit, external cooling
short name	
Definition:	

Example Value from ECLASS and ECLASS Advanced

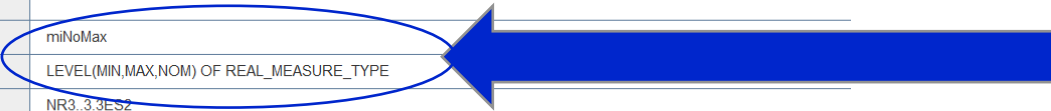
Change Text Replace Delete Copy

General Admin Attribute CR History Release

Value	BAB657
IRDI	0173-1#07-BAB657#003
eCI@ss v5 ID	BAB657001
Preferred Name	open circuit, external cooling
Short Name	
Definition	
Source of Definition	
Note to Definition	
Data Type	String
Value specification	Coded values
Exception	No

IEC 61360 Level Type

Code:	0112/2///61360_4#AAE022
Version:	001
Revision:	05
IRD:	0112/2///61360_4#AAE022#001
Preferred name:	outside diameter
Synonymous name:	
Symbol:	d _{out}
Synonymous symbol:	
Short name:	d_out
Definition:	value as specified by level (miNoMax) of the outside diameter of a component with a body of circular cross-section
Note:	
Remark:	
Primary unit:	m
Alternative units:	
Level:	miNoMax
Data type:	LEVEL(MIN,MAX,NOM) OF REAL_MEASURE_TYPE
Format:	NR3..3.ES2
Property constraint:	
Definition source:	
Value source:	
Property data element type:	NON_DEPENDENT_P_DET
Drawing:	
Formula:	
Value list code:	
Value list:	
DET class:	T03
Applicable classes:	0112/2///61360_4#AAA001 - component
Definition class:	0112/2///61360_4#AAA001
Code for unit:	0112/2///62720#UAA726 - metre
Codes for alternative units:	
Code for unit list:	



Example
Property with
Level Type
MIN, MAX
and NOM
from IEC
CDD

Administration shell

Serial number
Inverter current

i700E70

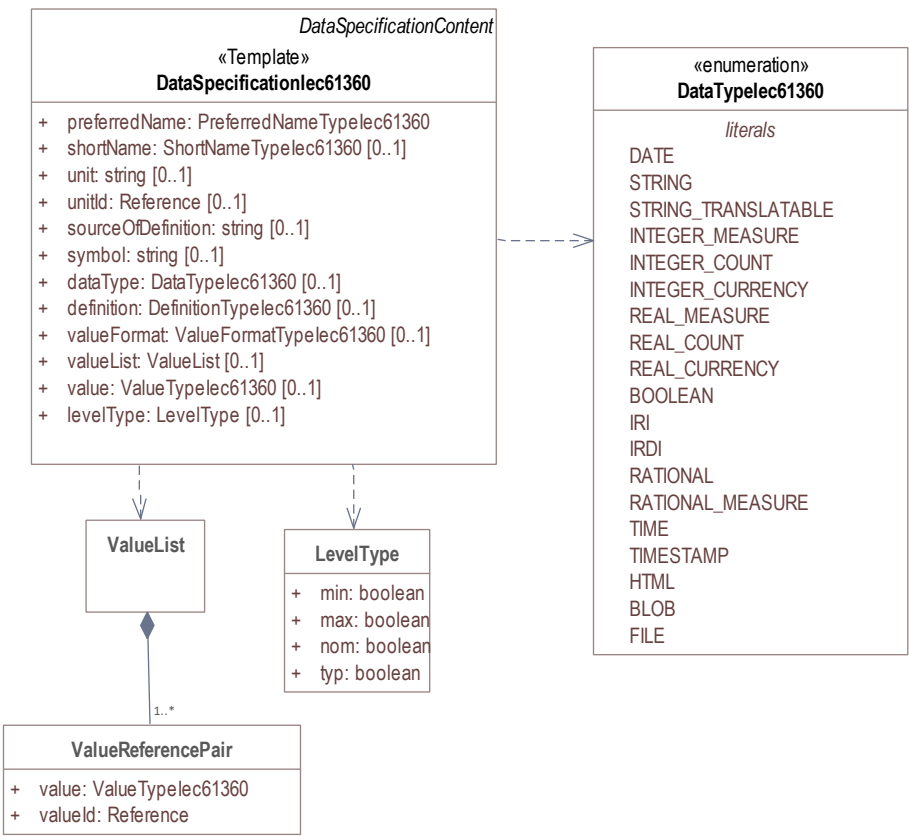
231231
0.02

Administration shell

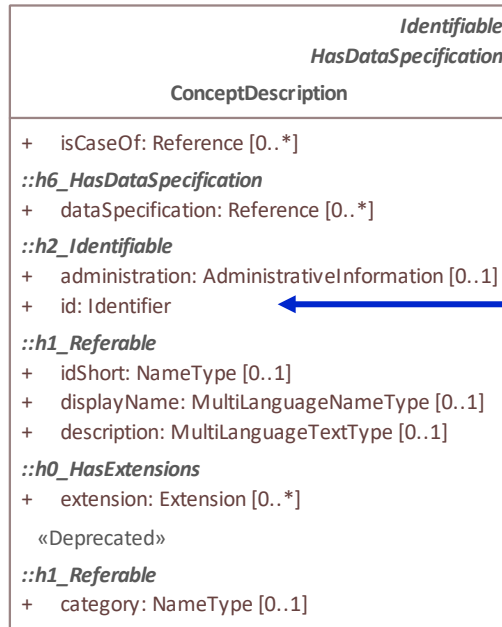
Serial number
Inverter current

● ● ● ● ● ● ● ●
● ● 0 ● ● 0 ● ●
● ● 1 ● ● 1 ● ●
● ● 0 ● ● ● ● ● ●
● ● 1 ● ● ● ● ● ●

IEC 61360



Property	0173-1#02-BAA120#008 Max. rotation speed
Data type	INTEGER_MEASURE
Unit of measure	1/min
Definition	Greatest possible rotation speed with wich the motor or feeding unit may be operated



Unique ID of property, value list or value defined via the data specification IEC62360



Note 1: IEC 61360 also requires a globally unique identifier for a concept description. This ID is not part of the data specification template. Instead, the *ConceptDescription/id* as inherited via *Identifiable* is used. The same applies to administrative information like the version and revision.

Note 2: *ConceptDescription/idShort* and *DataSpecificationIec61360/shortName* are very similar. However, in this case, *shortName* is explicitly added to the data specification.

Note 3: the same applies to *ConceptDescription/displayName* and *DataSpecificationIec61360/preferredName*.

Note 4: the same applies to *ConceptDescription/description* and *DataSpecificationIec61360/definition*.



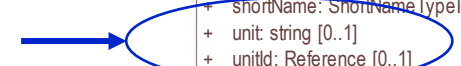
For Value Lists:

<i>DataSpecificationContent</i>	
«Template»	
DataSpecificationIec61360	
+ preferredName:	PreferredNameTypeIec61360
+ shortName:	ShortNameTypeIec61360 [0..1]
+ unit:	string [0..1]
+ unitId:	Reference [0..1]
+ sourceOfDefinition:	string [0..1]
+ symbol:	string [0..1]
+ dataType:	DataTypeIec61360 [0..1]
+ definition:	DefinitionTypeIec61360 [0..1]
+ valueFormat:	ValueFormatTypeIec61360 [0..1]
+ valueList:	ValueList [0..1]
+ value:	ValueTypeIec61360 [0..1]
+ levelType:	LevelType [0..1]

For measurable properties

(dataType = *_MEASURE)

<i>DataSpecificationContent</i>	
«Template»	
DataSpecificationIec61360	
+ preferredName:	PreferredNameTypeIec61360
+ shortName:	ShortNameTypeIec61360 [0..1]
+ unit:	string [0..1]
+ unitId:	Reference [0..1]
+ sourceOfDefinition:	string [0..1]
+ symbol:	string [0..1]
+ dataType:	DataTypeIec61360 [0..1]
+ definition:	DefinitionTypeIec61360 [0..1]
+ valueFormat:	ValueFormatTypeIec61360 [0..1]
+ valueList:	ValueList [0..1]
+ value:	ValueTypeIec61360 [0..1]
+ levelType:	LevelType [0..1]



For values (within value lists)

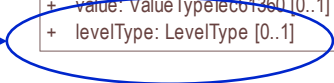
<i>DataSpecificationContent</i>	
«Template»	
DataSpecificationIec61360	
+ preferredName:	PreferredNameTypeIec61360
+ shortName:	ShortNameTypeIec61360 [0..1]
+ unit:	string [0..1]
+ unitId:	Reference [0..1]
+ sourceOfDefinition:	string [0..1]
+ symbol:	string [0..1]
+ dataType:	DataTypeIec61360 [0..1]
+ definition:	DefinitionTypeIec61360 [0..1]
+ valueFormat:	ValueFormatTypeIec61360 [0..1]
+ valueList:	ValueList [0..1]
+ value:	ValueTypeIec61360 [0..1]
+ levelType:	LevelType [0..1]





<i>DataSpecificationContent</i> «Template» DataSpecificationIec61360
+ preferredName: PreferredNameTypeIec61360
+ shortName: ShortNameTypeIec61360 [0..1]
+ unit: string [0..1]
+ unitId: Reference [0..1]
+ sourceOfDefinition: string [0..1]
+ symbol: string [0..1]
+ dataType: DataTypeIec61360 [0..1]
+ definition: DefinitionTypeIec61360 [0..1]
+ valueFormat: ValueFormatTypeIec61360 [0..1]
+ valueList: ValueList [0..1]
+ value: ValueIec61360 [0..1]
+ levelType: LevelType [0..1]

For properties with Level Type



<i>DataElement</i> «Experimental» Range
+ valueType: DataTypeDefXsd
+ min: ValueDataType [0..1]
+ max: ValueDataType [0..1]

Use level type MIN and MAX for submodel element „Range“ (see Part 1)

It is not recommended to use standardized properties with level types except those defining ranges.

«enumeration» DataTypeiec61360
<i>literals</i>
DATE STRING STRING_TRANSLATABLE INTEGER_MEASURE INTEGER_COUNT INTEGER_CURRENCY REAL_MEASURE REAL_COUNT REAL_CURRENCY BOOLEAN IRI IRDI RATIONAL RATIONAL_MEASURE TIME TIMESTAMP HTML BLOB FILE

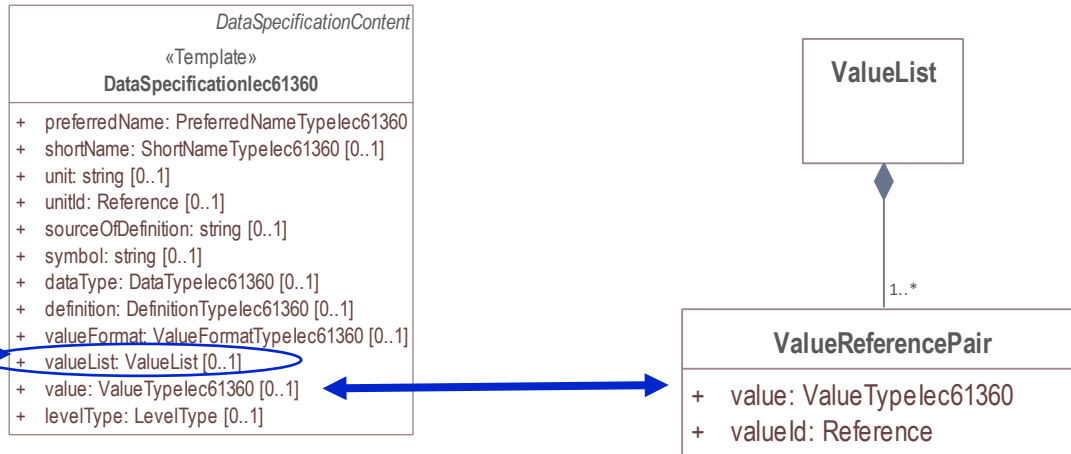
Data Type IEC 61360	xsd Value Type ³	Example Values IEC 61360 ⁴
DATE	xs:date	1979-01-15
STRING	xs:string	"DN 700" "10 Mbps"
STRING_TRANSLATABLE	Mapped to <i>MultiLanguageProperty</i> , i.e. type <i>MultiLanguageText</i> Note: for details, please see Part 1 of the document series "Details of the Asset Administration Shell".	
INTEGER_MEASURE	xs:integer	1 10 111
INTEGER_COUNT	xs:integer	1 10 111
INTEGER_CURRENCY	xs:integer	1 10 111
REAL_MEASURE	xs:double or xs:float (depending on needed)	1.5 102.35

Note: Data Type FILE is not explicitly supported by IEC61360 but used for data elements File

Value List



For Value Lists:



The valueId references a value

Property	02-BAE122 Cooling type
short name	-
Format	STRING
Definition:	Summary of various types of cooling, for use as search criteria that limit a selection
Values:	<ul style="list-style-type: none"> 0173-1#07-BAB649#001 - Air-air heat exchanger 0173-1#07-BAB650#001 - Air-water heat exchanger 0173-1#07-BAB592#001 - alien 0173-1#07-BAB611#001 - closed, external air-cooling 0173-1#07-BAB610#001 - closed, internal air-cooling 0173-1#07-BAB591#003 - free cooling 0173-1#07-BAB702#003 - Heat exchanger against other cooling medium 0173-1#07-BAB657#003 - open circuit, external cooling 0173-1#07-BAB656#003 - open circuit, internal cooling 0173-1#07-BAB535#003 - other form of cooling with primary air coolant 0173-1#07-BAB536#003 - other primary non-air coolant 0173-1#07-BAB674#003 - self

Level Type



LevelType
+ min: boolean
+ max: boolean
+ nom: boolean
+ typ: boolean

Class:	LevelType		
Explanation:	Value represented by up to four variants of a numeric value in a specific role: MIN, NOM, TYP, and MAX. True means that the value is available, false means the value is not available.		
	Note: for details, please refer to [IEC61360-1], LEVEL_TYPE		
	EXAMPLE from [IEC61360-1]: in case of a property which is of the LEVEL_TYPE min/max - Note: for details, please refer to [IEC61360-1], LEVEL_TYPE		
Inherits from:	DataSpecificationContent		
Attribute	Explanation	Type	Card.
min	Minimum of the value	boolean	1
nom	Nominal value (value as designated)	boolean	1
typ	Value as typically present	boolean	1
max	Maximum of the value	boolean	1

<i>DataElement</i>
«Experimental» Range
+ valueType: DataTypeDefXsd
+ min: ValueDataType [0..1]
+ max: ValueDataType [0..1]





0 0
1 1
0 0
1 1

Create your first digital twin

AASX Package Explorer

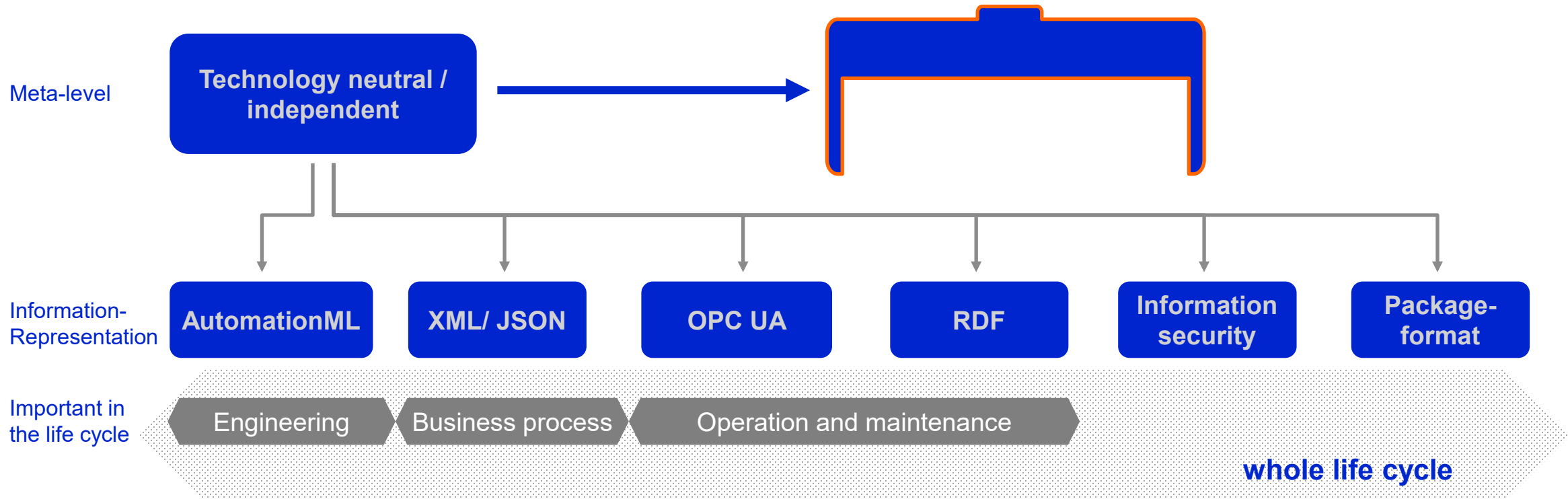
The screenshot shows the AASX Package Explorer V3RC02 interface. The main window displays a tree view of assets for a "ServoDriveCompactConverter". The selected asset is "TechnicalData", which is expanded to show "GeneralInformation", "ProductClassifications", "TechnicalProperties", "FurtherInformation", and "Certification". The "Certification" section is further expanded to show properties like "CertificationPresent", "Certificate_UL508C", "Certificate_EN61800_5_1", and "Certificate_EN61800_3".

The right-hand pane shows the "Technical Data" for the "Compact Converter" by Bosch Rexroth AG. It includes a logo and a table of properties:

Property	Semantics	Value
GeneralData		
ProtectionTypeOverall	0173-1#02-BAG975#012	IP20
Degree of contamination	http://boschrexroth.com/cd/Dc2	
Type of cooling	http://boschrexroth.com/cd/Cc	Forced Ventilation
PerformanceData		
Continuous current	0173-1#02-BAB295#006	4.4 A
Maximum current	0173-1#02-AAF853#003	13 A A
Mains connection voltage 1 AC	http://boschrexroth.com/cd/M	110 .. 230 V
Mains connection voltage 3 AC	http://boschrexroth.com/cd/M	110 .. 230 V
Tolerance	0173-1#02-AAV196#002	10 %
Frequency	0173-1#02-BAE130#007	50 .. 60
Frequency tolerance	0173-1#02-AAV198#002	2 %
Continuous current mains input	http://boschrexroth.com/cd/Cc	4.5 A
Power dependency from the su	http://boschrexroth.com/cd/Pc	1 % power reduction per 4 V
Power dependency from the su		No power increase

At the bottom of the interface, there is a status bar showing "0 bytes", "No errors", and "Clear Log..." buttons.

<https://github.com/admin-shell-io/aasx-package-explorer>



Serializations/Formats

- 7 Mappings to Data Formats to Share I4.0-Compliant Information (normative)
 - 7.1 General
 - 7.2 General Rules
 - 7.2.1 Introduction
 - 7.2.2 Encoding
 - 7.2.3 Serialization of Values of Type "Reference"
 - 7.2.4 Semantic Identifiers for Metamodel and Data Specifications
 - 7.2.5 Embedded Data Specifications
 - 7.3 XML
 - 7.4 JSON
 - 7.5 RDF
 - 7.6 AutomationML
 - 7.7 OPC UA

README.md

XML

Extensible Markup Language (XML) is a popular serialization format for data exchange and storage.

While there are many possibilities to represent a model of an Asset Administration Shell in XML, we provide our "official" definition (XSD) to foment interoperability between different tools and systems.

Below we explain in more detail how our schema is constructed, point the user to the examples and finally give some background on our particular schema design.

Top-Level Structure

The root element of our XML is an XML element representing the instance of `Environment`. This environment contains the corresponding to all `Identifiable` classes:

- `AssetAdministrationShell`'s,
- `Submodel`'s, and
- `ConceptDescription`'s.

To simplify exploration of the XML data, identifiable instances are only available at the level of the `Environment`.

We now continue to see how to serialize the instances and their properties.

Mapping Rules

Building blocks of an XML document include only `XML elements`, `XML attributes` and `text enclosed in an element`. XML elements can have children elements. Using these building blocks, we map an AAS model to XML.

UML Property to XML Element

Before we look into how to represent instances of classes, let us start bottom-up and see first how individual properties are represented.

We represent each property of a class with an XML element whose name corresponds to the property name in camel-case where all abbreviations are left as capitalized (`dataSpecificationIec61360` instead of `dataSpecificationIec61360s`).

It is common in UML to use singular form for aggregations, which is the case for the meta-model. In the code, where plural form for sequences is common. Since the naming of XML elements has direct influence on the properties in plural form diverging from the name in the meta-model. For example, `submodel` is used for a sequence of `Submodel` instances.

- json
- rdf
- xmi
- xml
- yaml
- .gitignore
- InstallSchemaValidation.ps1
- Validate.ps1

<https://github.com/admin-shell-io/aas-specs/tree/master/schemas>

Note for Experts: Mapping Rules and Schema for xml, JSON and rdf as well as examples not part of specification any longer → now part of open source project admin-shell-io/aas-specs

Note for Experts: Formats like OPC UA or AutomationML are maintained in OPC Foundation and Automation e.V.

Note: for data specifications the embedded approach is used

Note: see Readme files for different mappings to XML, JSON and RDF

Open Source Support



admin-shell-io by IDTA
Industrial Digital Twin Association e.V.
<https://idtwin.org/>

<https://github.com/orgs/admin-shell-io/>

Note: specifications maintained in admin-shell-io

ECLIPSE FOUNDATION Projects Working Groups

Home / Projects / Eclipse Digital Twin / Governance

Eclipse Digital Twin

Overview Downloads Who's Involved Developer Resources **Governance** Contact Us

Scope:
The Eclipse Digital Twin Top-Level Project supports projects at the Eclipse Foundation focusing on the implementation of solutions, prototypes and supporting software of digital twin technology .

The envisioned efforts include the following areas:

- Modelling and building digital twins based on open standards and technologies
- Modelling and consuming of existing and new open standards for the information provided via digital twins (dictionaries and semantic models/ontologies) components and modules for digital twins
- Infrastructural components for developing and operating digital twins
- Graphical User Interfaces for visualizing and interacting with digital twins
- Backend adapters for gathering data provided via digital twins in standardized formats
- Connection of digital twins with existing semantic dictionaries and ontologies
- Usage of digital twins in federated infrastructures
- Support of static (master data), dynamic (runtime) and behavioural data across the complete life cycle of an asset represented by a digital twin
- Lifecycle Management of digital twins
- Support of different development, testing, deployment, and operation strategies of digital twins
- Integration of digital twins with other technologies
- Development examples and demonstrators of digital twins and tools

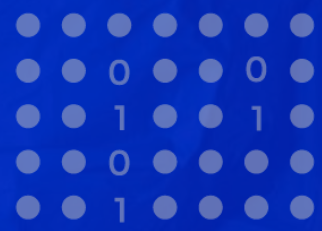
RELATED PROJECTS

Project Hierarchy:

- » Eclipse Digital Twin
 - » Eclipse AAS Model for Java
 - » Eclipse AAS Web Client
 - » Eclipse AASX Package Explorer
 - » Eclipse BaSyx™
 - » Eclipse Semantic Modeling Framework
 - » Eclipse Service Lifecycle Management

Status: September 2023

<https://projects.eclipse.org/projects/dt/>



Still Questions?

Questions and Answers



Search or jump to... Pull requests Issues Marketplace Explore

admin-shell-io / questions-and-answers Public Edit Pins Unwatch

Code Issues 13 Pull requests 1 Discussions Actions Projects Wiki S

master 6 branches 0 tags

StenGruener	Update README.md
AASBOK	Update README.md
Examples	isCaseOf example
reading-guide	2 and 3 fixes
README.md	Update README.md (#74)



Recommended documents

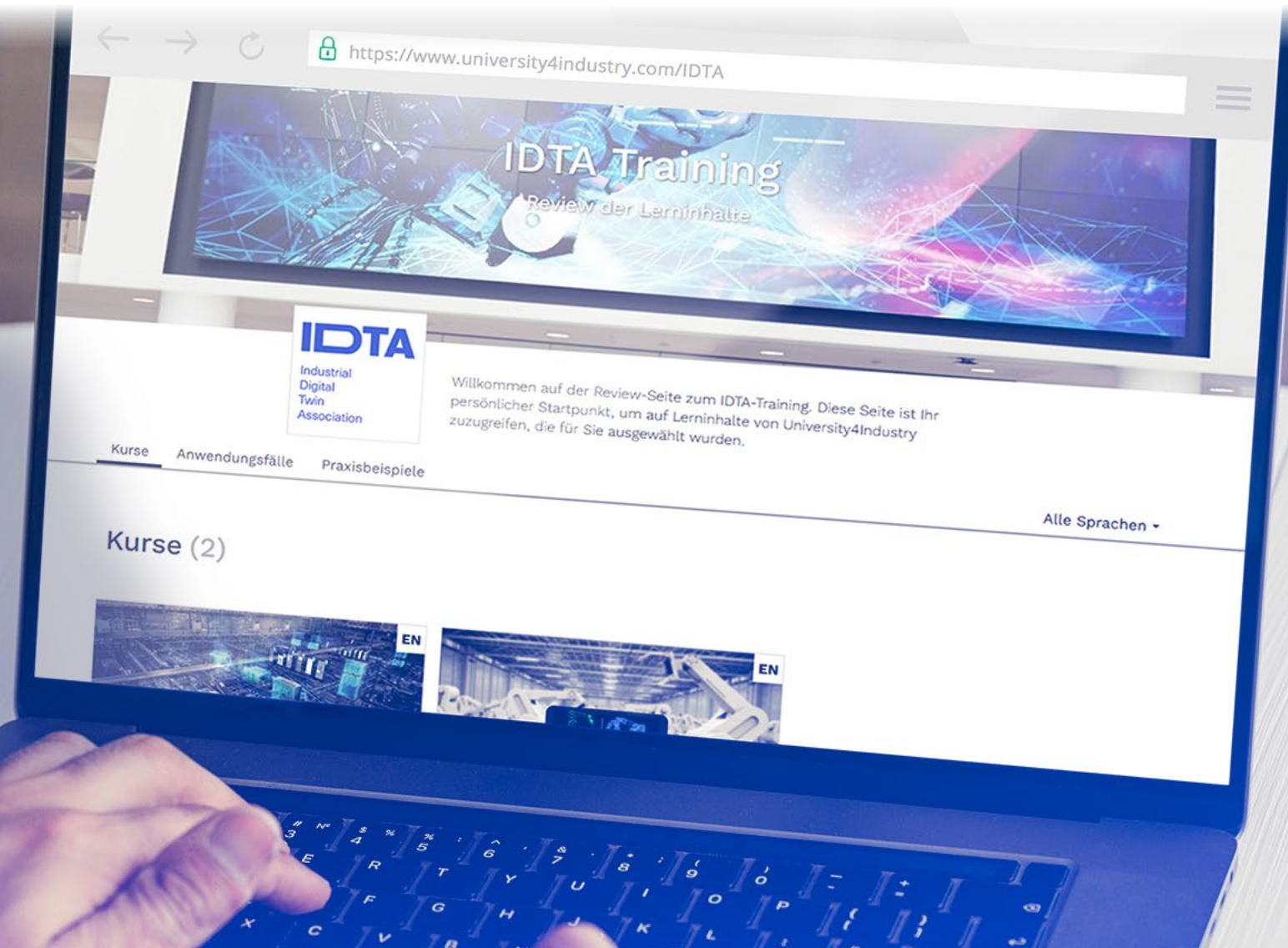
For this reading guide the documents have been sorted by interest groups rather than topics. In some cases, only specific pages or sections are recommended reading material.


- **Where to start:** If you have never heard of the AAS
- **For the generally interested reader:** If you want to learn more about the subject
- **For decision makers:** If you are interested in the business side of I4.0
- **For software developers and architects:** If you want to know how to create software for the AAS
- **For users of the AAS and domain experts:** If you are interested in using the AAS for specific tasks
- **Security and AI:** If you want to deep dive into these special topics.

README.md

Asset Administration Shell Frequently Asked Questions List

<https://github.com/admin-shell-io/questions-and-answers>



 www.u4i.io/IDTA

 Let's go!



Connect on

www.linkedin.com/in/birgit-boss/

Dr. Birgit Boss

Robert Bosch GmbH, Bosch Connected Industry

- Board member of the Industrial Digital Twin Association (IDTA)
- Chair of the Working Group “Open Technology” and its Working Stream “Specifications of the Asset Administration Shell”
- Chair of the Working Group “Semantic Layer including Digital Twins” of Catena-X
- PMC member of the Eclipse Digital Twin Top Level Project

