European Network of
Transmission System Operators
for Electricity

OBJECT REGISTRY PROFILE SPECIFICATION

2021-12-01

SOC APPROVED
VERSION 2.0

Copyright notice:

This document is maintained by the ENTSO-E CIM EG. Comments or remarks are to be provided at cim@entsoe.eu

NOTE CONCERNING WORDING USED IN THIS DOCUMENT

The force of the following words is modified by the requirement level of the document in which they are used.

- SHALL: This word, or the terms “REQUIRED” or “MUST”, means that the definition is an absolute requirement of the specification.
- SHALL NOT: This phrase, or the phrase “MUST NOT”, means that the definition is an absolute prohibition of the specification.
- SHOULD: This word, or the adjective “RECOMMENDED”, means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- SHOULD NOT: This phrase, or the phrase “NOT RECOMMENDED”, means that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.
- MAY: This word, or the adjective “OPTIONAL”, means that an item is truly optional.

33

Revision History

Version	Release	Date	Paragraph	Comments
1	0	2021-10-12		For CIM EG review
2	0	2021-12-01		SOC approved.

CONTENTS

List of figures

List of tables

76

1 Introduction

The object registry profile enables and exchange of different codes and names that relate to elements in the model.

2 Application profile specification

2.1 Version information

The content is generated from UML model file CGMES30v25_501-20v01_HeaderMetaData-10v08_NC20v57.eap.

This edition is based on the IEC 61970 UML version 'IEC61970CIM17v40', dated '2020-08-24'.

- Title: Object Registry vocabulary
- Keyword: OR
- Description: This vocabulary is describing the object registry profile.
- Version IRI: http://entsoe.eu/ns/CIM/ObjectRegistry-EU/2.0
- Version info: 2.0.0
- Prior version:
- Conforms to: urn:iso:std:iec:61970-600-2:ed-1|urn:iso:std:iec:61970-301:ed-7:amd1|file://iec61970cim17v40_iec61968cim13v13a_iec62325cim03v17a.eap|urn:iso:std:iec:61970-401:draft:ed-1|urn:iso:std:iec:61970-501:draft:ed-2|file://CGMES-30v25_501-20v01.eap
- Identifier: urn:uuid:14166b65-abaa-4611-b466-34975c15c27d

2.2 Constraints naming convention

The naming of the rules shall not be used for machine processing. The rule names are just a string. The naming convention of the constraints is as follows.

"{rule.Type}:{rule.Standard}:{rule.Profile}:{rule.Property}:{rule.Name}"

where

rule.Type: C – for constraint; R – for requirement

rule.Standard: the number of the standard e.g. 301 for 61970-301, 456 for 61970-456, 13 for 61968-13. 61970-600 specific constraints refer to 600 although they are related to one or combination of the 61970-450 series profiles. For NC profiles, NC is used.

rule.Profile: the abbreviation of the profile, e.g. TP for Topology profile. If set to "ALL" the constraint is applicable to all IEC 61970-600 profiles.

rule.Property: for UML classes, the name of the class, for attributes and associations, the name of the class and attribute or association end, e.g. EnergyConsumer, IdentifiedObject.name, etc. If set to "NA" the property is not applicable to a specific UML element.

rule.Name: the name of the rule. It is unique for the same property.

Example: C:600:ALL:IdentifiedObject.name:stringLength

2.3 Profile constraints

This clause defines requirements and constraints that shall be fulfilled by applications that conform to this document.

This document is the master for rules and constraints tagged "NC". For the sake of self-containment, the list below also includes a copy of the relevant rules from IEC 61970-452, tagged "452".

- C:452:ALL:NA:datatypes

  According to 61970-501, datatypes are not exchanged in the instance data. The UnitMultiplier is 1 in cases none value is specified in the profile.

- R:452:ALL:NA:exchange

  Optional and required attributes and associations must be imported and exported if they are in the model file prior to import.

- R:452:ALL:NA:exchange1

  If an optional attribute does not exist in the imported file, it does not have to be exported in case exactly the same data set is exported, i.e. the tool is not obliged to automatically provide this attribute. If the export is resulting from an action by the user performed after the import, e.g. data processing or model update the export can contain optional attributes.

- R:452:ALL:NA:exchange2

  In most of the profiles the selection of optional and required attributes is made so as to ensure a minimum set of required attributes without which the exchange does not fulfil its basic purpose. Business processes governing different exchanges can require mandatory exchange of certain optional attributes or associations. Optional and required attributes and associations shall therefore be supported by applications which claim conformance with certain functionalities of the IEC 61970-452. This provides flexibility for the business processes to adapt to different business requirements and base the exchanges on IEC 61970-452 compliant applications.

- R:452:ALL:NA:exchange3

  An exporter may, at his or her discretion, produce a serialization containing additional class data described by the CIM Schema but not required by this document provided these data adhere to the conventions established in Clause 5.

- R:452:ALL:NA:exchange4

  From the standpoint of the model import used by a data recipient, the document describes a subset of the CIM that importing software shall be able to interpret in order to import exported models. Data providers are free to exceed the minimum requirements described herein as long as their resulting data files are compliant with the CIM Schema and the conventions established in Clause 5. The document, therefore, describes additional classes and class data that, although not required, exporters will, in all likelihood, choose to include in their data files. The additional classes and data are labelled as required (cardinality 1..1) or as optional (cardinality 0..1) to distinguish them from their required counterparts. Please note, however, that data importers could potentially receive data containing instances of any and all classes described by the CIM Schema.

- R:452:ALL:NA:cardinality

The cardinality defined in the CIM model shall be followed, unless a more restrictive cardinality is explicitly defined in this document. For instance, the cardinality on the association between VoltageLevel and BaseVoltage indicates that a VoltageLevel shall be associated with one and only one BaseVoltage, but a BaseVoltage can be associated with zero to many VoltageLevels.

- R:452:ALL:NA:associations

Associations between classes referenced in this document and classes not referenced here are not required regardless of cardinality.

- R:452:ALL:IdentifiedObject.name:rule

The attribute “name” inherited by many classes from the abstract class IdentifiedObject is not required to be unique. It must be a human readable identifier without additional embedded information that would need to be parsed. The attribute is used for purposes such as User Interface and data exchange debugging. The MRID defined in the data exchange format is the only unique and persistent identifier used for this data exchange. The attribute IdentifiedObject.name is, however, always required for CoreEquipment profile and Short Circuit profile.

- R:452:ALL:IdentifiedObject.description:rule

The attribute “description” inherited by many classes from the abstract class IdentifiedObject must contain human readable text without additional embedded information that would need to be parsed.

- R:452:ALL:NA:uniqueIdentifier

All IdentifiedObject-s shall have a persistent and globally unique identifier (Master Resource Identifier - mRID).

- R:452:ALL:NA:unitMultiplier

For exchange of attributes defined using CIM Data Types (ActivePower, Susceptance, etc.) a unit multiplier of 1 is used if the UnitMultiplier specified in this document is “none”.

- C:452:ALL:IdentifiedObject.name:stringLength

The string IdentifiedObject.name has a maximum of 128 characters.

- C:452:ALL:IdentifiedObject.description:stringLength

The string IdentifiedObject.description is maximum 256 characters.

- C:452:ALL:NA:float

An attribute that is defined as float (e.g. has a type Float or a type which is a Datatype with .value attribute of type Float) shall support ISO/IEC 60559:2020 for floating-point arithmetic using single precision floating point. A single precision float supports 7 significant digits where the significant digits are described as an integer, or a decimal number with 6 decimal digits. Two float values are equal when the significant with 7 digits are identical, e.g. 1234567 is equal 1.234567E6 and so are 1.2345678 and 1.234567E0.

2.4 Metadata

ENTSO-E agreed to extend the header and metadata definitions by IEC 61970-552 Ed2. This new header definitions rely on W3C recommendations which are used worldwide and are positively recognised by the European Commission. The new definitions of the header mainly use Provenance ontology (PROV-O), Time Ontology and Data Catalog Vocabulary (DCAT). The global new header applicable for this profile is included in the metadata and document header specification document.

The header vocabulary contains all attributes defined in IEC 61970-552. This is done only for the purpose of having one vocabulary for header and to ensure transition for data exchanges that are using IEC 61970-552:2016 header. This profile does not use IEC 61970-552:2016 header attributes and relies only on the extended attributes.

2.4.1 Constraints

The identification of the constraints related to the metadata follows the same convention for naming of the constraints as for profile constraints.

- R:NC:ALL:wasAttributedTo:usage

  The prov:wasAttributedTo should normally be the “X” EIC code of the actor (prov:Agent).

2.4.2 Reference metadata

The header defined for this profile requires availability of a set of reference metadata. For instance, the attribute prov:wasGeneratedBy requires a reference to an activity which produced the model or the related process. The activities are defined as reference metadata and their identifiers are referenced from the header to enable the receiving entity to retrieve the “static” (reference) information that is not modified frequently. This approach imposes a requirement that both the sending entity and the receiving entity have access to a unique version of the reference metadata. Therefore, each business process shall define which reference metadata is used and where it is located.

3 Detailed Profile Specification

3.1 General

This package contains the equipment object registry profile.

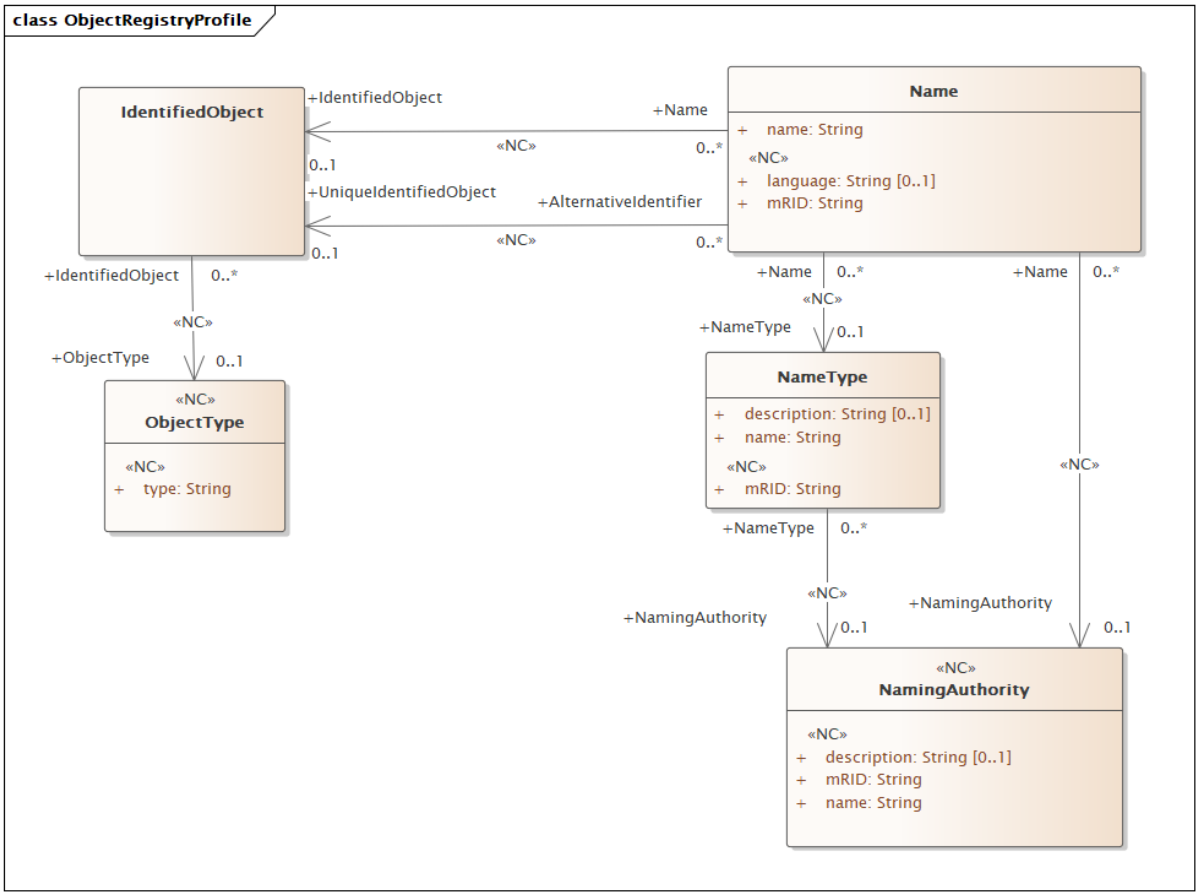

Figure 1 – Class diagram ObjectRegistryProfile::ObjectRegistryProfile

Figure 1: The diagram contains main classes related to the object registry profile.

3.2 IdentifiedObject root class

This is a root class to provide common identification for all classes needing identification and naming attributes.
Table 1 shows all association ends of IdentifiedObject with other classes.

Table 1 – Association ends of ObjectRegistryProfile::IdentifiedObject with other classes

mult from	name	mult to	type	description
0..*	ObjectType	0..1	ObjectType	(NC) The object type of the IdentifiedObject.

3.3 Name root class

The Name class provides the means to define any number of human readable names for an object. A name is not to be used for defining inter-object relationships. For inter-object relationships instead use the object identification 'mRID'.
Table 2 shows all attributes of Name.

Table 2 – Attributes of ObjectRegistryProfile::Name

name	mult	type	description
name	1..1	String	Any free text that used as a name or alternative identifier of the object.

name	mult	type	description
language	0..1	String	(NC) Shall be specified as an IETF BCP 47 language tag (e.g. en-US). Applies to the Name.name attribute. IETF language tags combine subtags from other standards such as ISO 639, ISO 15924, ISO 3166-1, and UN M.49. The tag structure has been standardized by the IETF in Best Current Practice (BCP) 47; the subtags are maintained by the IANA Language Subtag Registry.
mRID	1..1	String	(NC) Master resource identifier issued by a model authority. The mRID is unique within an exchange context. Global uniqueness is easily achieved by using a UUID, as specified in RFC 4122, for the mRID. The use of UUID is strongly recommended. For CIMXML data files in RDF syntax conforming to IEC 61970-552, the mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements.

Table 3 shows all association ends of Name with other classes.

Table 3 – Association ends of ObjectRegistryProfile::Name with other classes

mult from	name	mult to	type	description
0..*	UniqueIdentifiedObject	0..1	IdentifiedObject	(NC) Identified object that this alternative identifier designates.
0..*	IdentifiedObject	0..1	IdentifiedObject	(NC) Identified object that this name designates.
0..*	NameType	0..1	NameType	(NC) Type of this name.
0..*	NamingAuthority	0..1	NamingAuthority	(NC) Authority responsible for managing this name.

3.4 NameType root class

Type of name. Possible values for attribute 'name' are implementation dependent but standard profiles may specify types. An enterprise may have multiple IT systems each having its own local name for the same object, e.g. a planning system may have different names from an EMS. An object may also have different names within the same IT system, e.g. localName as defined in CIM version 14. The definition from CIM14 is:
The localName is a human readable name of the object. It is a free text name local to a node in a naming hierarchy similar to a file directory structure. A power system related naming hierarchy may be: Substation, VoltageLevel, Equipment etc. Children of the same parent in such a hierarchy have names that typically are unique among them.
Table 4 shows all attributes of NameType.

Table 4 – Attributes of ObjectRegistryProfile::NameType

name	mult	type	description
name	1..1	String	Name of the name type.
description	0..1	String	Description of the name type.
mRID	1..1	String	(NC) Master resource identifier issued by a model authority. The mRID is unique within an exchange context. Global uniqueness is easily achieved by using a UUID, as specified in RFC 4122, for the mRID. The use of UUID is strongly recommended.

name	mult	type	description
			For CIMXML data files in RDF syntax conforming to IEC 61970-552, the mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements.

Table 5 shows all association ends of NameType with other classes.

Table 5 – Association ends of ObjectRegistryProfile::NameType with other classes

mult from	name	mult to	type	description
0..*	NamingAuthority	0..1	NamingAuthority	(NC) Authority responsible for managing this name type.

3.5 (NC) NamingAuthority root class

Authority responsible for creation and management of names of a given name type and/or name; typically an organization or an enterprise system.
Table 6 shows all attributes of NamingAuthority.

Table 6 – Attributes of ObjectRegistryProfile::NamingAuthority

name	mult	type	description
description	0..1	String	(NC) Description of the name authority.
mRID	1..1	String	(NC) Master resource identifier issued by a model authority. The mRID is unique within an exchange context. Global uniqueness is easily achieved by using a UUID, as specified in RFC 4122, for the mRID. The use of UUID is strongly recommended. For CIMXML data files in RDF syntax conforming to IEC 61970-552, the mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements.
name	1..1	String	(NC) Name of the name authority.

3.6 (NC) ObjectType root class

Identifies the specialised type of an object when the instance object is serialised using a generalised class. It may be useful when the object type is not otherwise included in the exchange. For example, a Meter may be serialised as an EndDevice in message exchanges and need to have the ObjectType.type be specified as 'Meter' to provide context to the message receiver.
Table 7 shows all attributes of ObjectType.

Table 7 – Attributes of ObjectRegistryProfile::ObjectType

name	mult	type	description
type	1..1	String	(NC) The specialised type of an object when the instance object is serialised using a generalised class. For example, a Meter being serialised as an EndDevice in a message exchange should have the type attribute specified as 'Meter'.

3.7 Date primitive

Date as "yyyy-mm-dd", which conforms with ISO 8601. UTC time zone is specified as "yyyy-mm-ddZ". A local timezone relative UTC is specified as "yyyy-mm-dd(+/-)hh:mm".

3.8 DateTime primitive

Date and time as "yyyy-mm-ddThh:mm:ss.sss", which conforms with ISO 8601. UTC time zone is specified as "yyyy-mm-ddThh:mm:ss.sssZ". A local timezone relative UTC is specified as "yyyy-mm-ddThh:mm:ss.sss-hh:mm". The second component (shown here as "ss.sss") could have any number of digits in its fractional part to allow any kind of precision beyond seconds.

3.9 (profcim) IRI primitive

An IRI (Internationalized Resource Identifier) within an RDF graph is a Unicode string that conforms to the syntax defined in RFC 3987.
The primitive is serialized as rdf:resource in RDFXML.
IRIs in the RDF abstract syntax must be absolute, and may contain a fragment identifier.
IRI equality: Two IRIs are equal if and only if they are equivalent under Simple String Comparison according to section 5.1 of [RFC3987]. Further normalization must not be performed when comparing IRIs for equality.
IRIs are a generalization of URIs [RFC3986] that permits a wider range of Unicode characters. Every absolute URI and URL is an IRI, but not every IRI is an URI. When IRIs are used in operations that are only defined for URIs, they must first be converted according to the mapping defined in section 3.1 of [RFC3987]. A notable example is retrieval over the HTTP protocol. The mapping involves UTF-8 encoding of non-ASCII characters, %-encoding of octets not allowed in URIs, and Punycode-encoding of domain names.

3.10 String primitive

A string consisting of a sequence of characters. The character encoding is UTF-8. The string length is unspecified and unlimited.

3.11 (profcim) StringFixedLanguage primitive

A string consisting of a sequence of characters. The character encoding is UTF-8. The string length is unspecified and unlimited.
The primitive is serialized as literal without language support.

3.12 (profcim) StringIRI primitive

An IRI (Internationalized Resource Identifier) within an RDF graph is a Unicode string that conforms to the syntax defined in RFC 3987.
The primitive is serialized as literal without language support.
IRIs in the RDF abstract syntax must be absolute, and may contain a fragment identifier.
IRI equality: Two IRIs are equal if and only if they are equivalent under Simple String Comparison according to section 5.1 of [RFC3987]. Further normalization must not be performed when comparing IRIs for equality.
IRIs are a generalization of URIs [RFC3986] that permits a wider range of Unicode characters. Every absolute URI and URL is an IRI, but not every IRI is an URI. When IRIs are used in operations that are only defined for URIs, they must first be converted according to the mapping defined in section 3.1 of [RFC3987]. A notable example is retrieval over the HTTP protocol. The mapping involves UTF-8 encoding of non-ASCII characters, %-encoding of octets not allowed in URIs, and Punycode-encoding of domain names.

3.13 (profcim) URL primitive

A Uniform Resource Locator (URL), colloquially termed a web address, is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it. A URL is a specific type of Uniform Resource Identifier (URI), although many people use the two terms interchangeably.URLs occur most commonly to reference web pages (http), but are also used for file transfer (ftp), email (mailto), database access (JDBC), and many other applications.

Annex A (informative): Sample data

A.1 General

This Annex is designed to illustrate the profile by using fragments of sample data. It is not meant to be a complete set of examples covering all possibilities of using the profile. Defining a complete set of test data is considered a separate activity to be performed for the purpose of setting up interoperability testing and conformity related to this profile.

A.2 Object Registry profile

```
<cim:Name rdf:ID="_fd1919e8-b8f9-41d6-870e-785700665e4c">
  <cim:Name.name>43T-LV---LN320-W</cim:Name.name>
  <nc:Name.mRID>fd1919e8-b8f9-41d6-870e-785700665e4c</cim:IdentifiedObject.mRID>
  <nc:Name.IdentifiedObject rdf:resource="#_00f5b7fc-e6f4-435d-8826-35abbf388ec7" />
  <nc:Name.NameType rdf:resource="#_b025b353-1dbc-422f-88cf-d84d73d4371b" />
  <nc:Name.NamingAuthority rdf:resource #_f4ace05a-ab03-43f5-a39d-d65b838b6c11" />
</cim:Name>

<cim:NameType rdf:ID="_b025b353-1dbc-422f-88cf-d84d73d4371b ">
  <cim:NameType.name>EIC</cim:NameType.name>
  <nc:NameType.mRID>b025b353-1dbc-422f-88cf-d84d73d4371b</nc:NameType.mRID>
</cim:NameType >
<nc:NamingAuthority rdf:ID="_f4ace05a-ab03-43f5-a39d-d65b838b6c11">
  <cim:NamingAuthority.name>LIO</cim:NamingAuthority.name>
  <nc:NamingAuthority.mRID>f4ace05a-ab03-43f5-a39d-d65b838b6c11</nc:NamingAuthority.mRID>
</nc:NamingAuthority >
```