European Network of
Transmission System Operators
for Electricity

OBJECT REGISTRY PROFILE SPECIFICATION

2026-09-10

ICTC APPROVED
VERSION 3.0.0

Copyright notice:

This document is maintained by the ENTSO-E CIM WG. 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	Date	Paragraph	Comments
1.0.0	2021-10-12		For CIM EG review
2.0.0	2021-11-15		SOC approved.
2.1.0	2022-09-21		SOC approved.
2.2.0-alpha	2024-03-20		For CIM WG review.
2.2.1-alpha	2024-09-07		For CIM WG review.
3.0.0	2026-08-05		For CIM WG review.
3.0.0	2026-09-10		ICTC approval.

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 CIM17-2_CGMES31v01_PROF-20v03_NC25v25_MS10v12_DES10v01.qea.

This edition is based on the IEC 61970 UML version ‘’, dated ‘’.

- Title:
- Keyword:
- Description:
- Version IRI:
- Version info:
- Prior version:
- Conforms to:
- Identifier:

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 CSA profiles, CSA 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 "CSA". 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.

- R:NC:ALL:NA:serialization

The profiles are defined in the EnterpriseArchitect application and have multiple artifacts that describe them. The main artifacts are:

1) the EAP file (EnterpriseArchitect project file),
2) the profiles’ specification document and

3) the application profiles (RDFS and SHACL).

Due to the complexity of the profiles, there are various cross profile associations that, from profiling and profile maintenance point of view, it is not practical to include the complete inheritance structure in all profiles. If this is done the documentation provided for all profiles would also include duplicated information on the description of classes defined in other profiles. The following cases are often observed in profiles:

- Case 1: An association end refers to an abstract class
- Case 2: An abstract class (stereotyped with “Description”) has an association (direction to another class)
- Case 3: An abstract class (not stereotyped with “Description”) has an association (direction to another class)
- Case 4: An abstract class has attributes and subclasses are not in the profile

In all cases, the datasets shall only include the subtypes of the abstract classes with the related properties (i.e. association or attributes) defined in the profile. The information is taken from either canonical model or the profiles where complete (expected) inheritance structure for the related abstract class is described. SHACL based constraints include constraints only for the concrete classes that are subtypes of the abstract class in the profile, and this can be used to inform which are the concrete classes expected in a dataset that conforms to this profile.

It should be taken into account that this approach deviates from MVAL5 (IEC 61970-600-1:2021), which creates multiple inheritance at serialization. For instance, with this more explicit exchange the serialization of the association between abstract class Equipment and abstract class Circuit for a PowerTransformer will be serialized as follows:

- for association

```
<cim:PowerTransformer rdf:about="_c328f787-bc17-47ad-a59f-6ba7133340d0">
  <nc:Equipment.Circut rdf:resource="#_9ced16ac-d076-4ef9-a241-a998a579e77b"/>
</cim:PowerTransformer>
```

- for attribute

```
<cim:ACLineSegment rdf:about="_04f681aa-6999-4fb3-9775-acaa5eb7ceff">
  <cim:Equipment.inService>true</cim:Equipment.inService>
</cim:ACLineSegment>
```

The usage of rdf:ID or rdf:about depends on the stereotype of the class. rdf:about is used if the class has the stereotype “Description”.

An example of not allowed serialization, as the Equipment is an abstract class

```
<cim:Equipment rdf:about="_c328f787-bc17-47ad-a59f-6ba7133340d0">
  <nc:Equipment.Circut rdf:resource="#_9ced16ac-d076-4ef9-a241-a998a579e77b"/>
</cim:Equipment>
```

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:CSA:ALL:wasAttributedTo:usage

  The prov:wasAttributedTo should normally be the “X” EIC code of the actor or their URI (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.

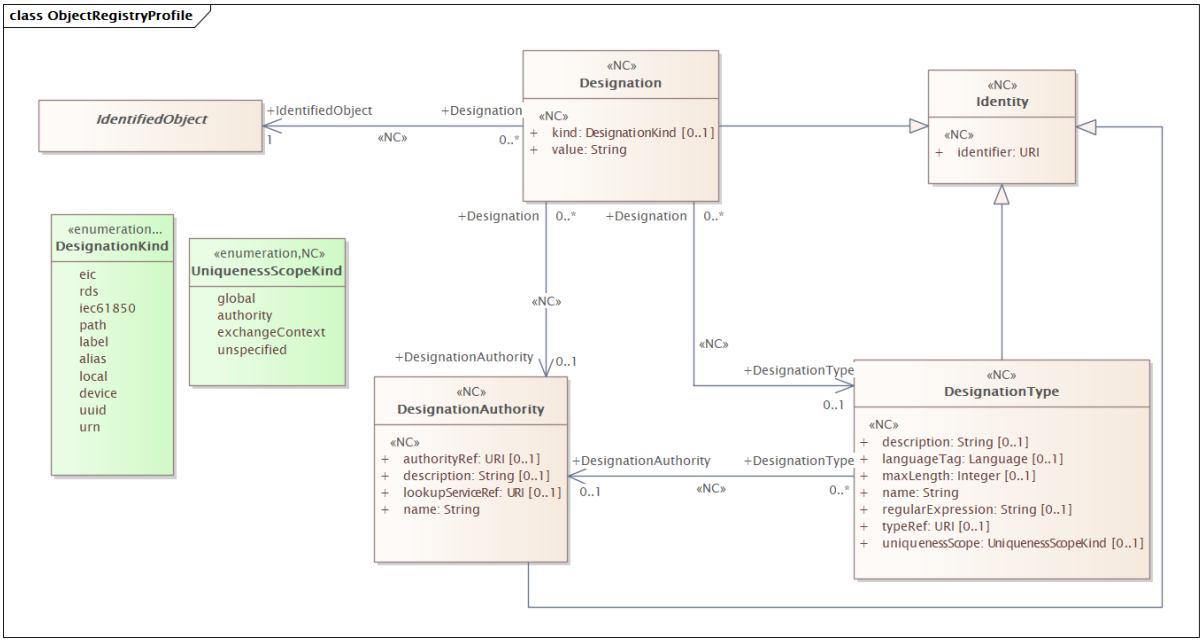

Figure 1 – Class diagram ObjectRegistryProfile::ObjectRegistryProfile

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

3.2 (abstract) IdentifiedObject root class

This is a root class to provide common identification for all classes needing identification and naming attributes.

3.3 (NC) DesignationKind enumeration

A standardised classification of designation values used to refer to an IdentifiedObject in different contexts. DesignationKind values are intended to be stable and directly usable in profiles and implementation guidance without requiring external reference data.
Table 1 shows all literals of DesignationKind.

Table 1 – Literals of ObjectRegistryProfile::DesignationKind

literal	value	description
eic		An Energy Identification Code designation, typically used in European energy market and interoperability contexts. Example: "10YNO-0--------C"
rds		A designation following the Reference Designation System (RDS) defined in IEC 81346, used to uniquely identify systems, functions, and equipment within engineering documentation and technical installations. RDS designations express the position or role of an object within a structured engineering hierarchy and are widely used in electrical engineering documentation, drawings, and plant documentation. Note: The Reference Designation System (RDS) provides a structured identification scheme where each segment represents an aspect of an object such as function ("="), product ("-"), or location ("+"). Example: "=N01+SA1-Q01"

literal	value	description
iec61850		A designation used in IEC 61850 contexts (e.g., IED name, logical node naming, object references, or related 61850 naming forms as specified by the profile). Example: "LD0/XCBR1.Pos.stVal"
path		A hierarchical designation constructed from a sequence of name segments representing containment, structure, or functional decomposition. Example: "SubstationA/Bay3/Breakder1"
label		A short designation primarily intended for local display, compact user interfaces, and single-line diagrams. A label is generally preferred to be short and readable rather than globally unique. Example: "T1", "Bay3"
alias		An alternative designation used by a system, project, or user community that differs from the default or preferred designation.
local		A designation used within a local naming scope (site, system, organisation, or sub-hierarchy) where meaning is derived from the local context. The name shall be unique inside the container, e.g. substation.
device		A designation used to identify a device, component, or object instance within a vendor system, device platform, or operational system. Such designations are typically assigned by manufacturers, device firmware, automation systems, metering systems, or enterprise applications (e.g., ERP, asset management, monitoring platforms) and are used for configuration, monitoring, or asset tracking. Note: Device designations are often defined within the scope of a particular vendor, system, or installation and are therefore typically unique only within that system context. Example: "IED_BAY3_01", "Meter_102394", "SAP_EQ_000457812"
uuid		A designation expressed as a Universally Unique Identifier (UUID) as defined in IETF RFC 4122 (and its successor RFC 9562). UUIDs are 128-bit identifiers designed to be globally unique without requiring a central registration authority. UUID designations are commonly used in software systems and data exchanges to provide stable identifiers across distributed systems. Note: A UUID is typically represented as a 36-character string consisting of hexadecimal digits grouped in five sections separated by hyphens. Example: "550e8400-e29b-41d4-a716-446655440000"
urn		A designation expressed as a Uniform Resource Name (URN), a persistent identifier defined using a registered namespace according to IETF RFC 8141. URN namespaces provide globally unique identifiers without requiring a resolvable URL and are commonly defined by standards bodies, industry communities, or organisations. Note: URN identifiers use the general structure: "urn:<namespace>:<identifier>" where

literal	value	description
		<namespace> identifies a registered namespace and <identifier> is the value assigned within that namespace. Example: Object Identifier (OID) "urn:oid:1.3.6.1.4.1.4555", UUID namespace "urn:uuid:550e8400-e29b-41d4-a716-446655440000", EIC namespace "urn:eic:10YNO-0--------C"

3.4 (NC) UniquenessScopeKind enumeration

Specifies the intended scope within which a class attribute value is unique.
Table 2 shows all literals of UniquenessScopeKind.

Table 2 – Literals of ObjectRegistryProfile::UniquenessScopeKind

literal	value	description
global		Intended to be globally unique (across organisations and exchanges).
authority		Unique within the scope of the associated authority.
exchangeContext		Unique only within the scope of a single exchange package/context (file set, model exchange).
unspecified		Uniqueness is not stated (do not assume uniqueness).

3.5 Integer primitive

An integer number. The range is unspecified and not limited.

3.6 String primitive

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

3.7 (European) URI primitive

URI is a string following the rules defined by the W3C/IETF URI Planning Interest Group in a set of RFCs of which one is RFC 3305.

3.8 (NC) Designation

Inheritance path = Identity
A designation is a string assigned to an IdentifiedObject to support recognition, presentation, search, and interoperability with external systems. A designation may be a human-readable label (e.g., display name, alias) or a system-issued code (e.g., EIC, IEC 61850 name, RDS reference designation). Designations can be typed and scoped by a designation authority.
Note:
Inter-object relationships in exchanged models shall reference the target object using its IdentifiedObject.mRID (or equivalent resolvable identifier), not by designation strings.
A designation value may be unique within a stated scope (e.g., globally, within an authority, or within an exchange context), but such uniqueness does not change the requirement to reference objects by mRID in relationships.
During transition and integration, multiple designations may be carried to support legacy identifiers, vendor/system naming, and user-facing labels.
Table 3 shows all attributes of Designation.

Table 3 – Attributes of ObjectRegistryProfile::Designation

name	mult	type	description
kind	0..1	DesignationKind	(NC) Classification indicating the general type or intended usage of the designation. Note: The classification allows systems and profiles to interpret designation values consistently. Additional constraints or specialisations may be defined through DesignationType.
value	1..1	String	(NC) The textual value of the designation assigned to an IdentifiedObject. Note: The value may represent a name, label, code, or identifier used to refer to the object in a particular context. Designations may be used for display, engineering documentation, system integration, or external identification. Object relationships in exchanged models shall reference objects using IdentifiedObject.mRID, not designation values.
identifier	1..1	URI	(NC) inherited from: Identity

Table 4 shows all association ends of Designation with other classes.

Table 4 – Association ends of ObjectRegistryProfile::Designation with other classes

mult from	name	mult to	type	description
0..*	IdentifiedObject	1..1	IdentifiedObject	(NC) The IdentifiedObject that is referred to in the Designation.
0..*	DesignationAuthority	0..1	DesignationAuthority	(NC) The authority responsible for issuing, assigning, or governing the designation.
0..*	DesignationType	0..1	DesignationType	(NC) Reference to a non-standard type specification that further classifies or constrains this designation.

3.9 (NC) DesignationAuthority

Inheritance path = Identity
An organisation, system, or community responsible for defining, issuing, or managing designations or designation types within a particular designation scheme or namespace.
Note:
A designation authority may define:
- designation schemes or namespaces
- designation type specifications
- rules for generating or validating designation values
Examples include standards organisations, market operators, manufacturers, utilities, or enterprise systems.
The designation authority may also provide a service or registry through which designation values or type definitions can be resolved or validated.
Example:
ENTSO-E
Designation authority responsible for issuing Energy Identification Codes (EIC).
GS1
Designation authority managing GS1 identifiers used for asset and product identification in Digital Product Passport and supply-chain contexts.
Equipment manufacturer

A vendor may act as designation authority for identifiers assigned to devices or assets within its monitoring or asset management systems.
Table 5 shows all attributes of DesignationAuthority.

Table 5 – Attributes of ObjectRegistryProfile::DesignationAuthority

name	mult	type	description
description	0..1	String	(NC) Textual description of the designation authority. Note: Provides additional information about the role, scope, or responsibilities of the authority managing the designation scheme.
name	1..1	String	(NC) Human-readable name of the designation authority. Note: Identifies the organisation, system, or community responsible for defining, issuing, or managing the designation scheme or designation types. Example: "ENTSO-E", "GS1", "Statnett Asset Registry"
lookupServiceRef	0..1	URI	(NC) URI reference to a service or endpoint that can be used to resolve, validate, or obtain additional information about designations issued by this authority. Note: The lookup service may provide information about designation values, identifier registries, or related metadata. Example: "https://id.gs1.org/"
authorityRef	0..1	URI	(NC) URI reference identifying the designation authority. Note: The URI may refer to an organisation identifier, registry entry, or other authoritative source describing the authority. Example: "https://www.gs1.org", "https://www.entsoe.eu/data/energy-identification-codes-eic/"
identifier	1..1	URI	(NC) inherited from: Identity

3.10 (NC) DesignationType

Inheritance path = Identity
A specification that defines an additional, non-standard designation type used to classify and constrain Designation values beyond the standardised DesignationKind enumeration. A DesignationType may be published by a designation authority and may provide validation and usage rules (e.g., maximum length, pattern, and intended uniqueness scope).
Note:
DesignationKind provides standardised categories that can be referenced directly in profiles (e.g., a diagram profile may require DesignationKind.local or DesignationKind.label).
DesignationType is used only when the designation required is not covered by the standardised DesignationKind list, or when a community needs to define additional constraints or sub-typing within a kind.
DesignationType is optional and should not be required for basic interoperability; consumers should be able to interpret Designation.kind without resolving external reference data.
If both are present, Designation.kind carries the standard category, while Designation.DesignationType carries the additional or more specific classification.
Table 6 shows all attributes of DesignationType.

Table 6 – Attributes of ObjectRegistryProfile::DesignationType

name	mult	type	description
languageTag	0..1	Language	(NC) Language tag identifying the language used for the name and description. Note: Language tags should follow BCP 47 (e.g., en, nb, en-GB).
description	0..1	String	(NC) Textual description of the designation type. Note: Explains the intended usage, scope, or rules associated with designations of this type.
uniquenessScope	0..1	UniquenessScopeKind	(NC) Intended scope within which designation value of this type are unique.
name	1..1	String	(NC) Human-readable name of the designation type. Note: Provides a short label describing the purpose or usage of the designation type. Example: "DiagramLabel", "AssetIdentifier", "SCADAName"
maxLength	0..1	Integer	(NC) Maximum allowed length of designation values of this type, expressed as number of characters.
regularExpression	0..1	String	(NC) Regular expression that a designation value shall conform to when this type is used.
typeRef	0..1	URI	(NC) URI reference to an external definition or specification describing the designation type. Note: The referenced resource may provide additional documentation, constraints, or governance information for the designation type. Example: https://energy.referencedata.eu/DesignationType/DiagramLabel
identifier	1..1	URI	(NC) inherited from: Identity

Table 7 shows all association ends of DesignationType with other classes.

Table 7 – Association ends of ObjectRegistryProfile::DesignationType with other classes

mult from	name	mult to	type	description
0..*	DesignationAuthority	0..1	DesignationAuthority	(NC) The authority that publishes or governs this type specification.

3.11 (NC) Identity root class

The Identity class represents the persistent identity of an object. It provides a globally unique and stable identifier used to support lifecycle management, change tracking, cross-system referencing, and model comparison.
Note: The identifier represents the enduring identity of the object and should remain stable throughout the object's lifecycle. In distributed systems this identifier is typically represented using a UUID to ensure global uniqueness.
The Identity class is intended to provide the stable identifier for an object, whereas classes such as IdentifiedObject and Designation provide human-readable names, labels, or external identifiers that may vary depending on context or system usage.
Table 8 shows all attributes of Identity.

372 **Table 8 – Attributes of ObjectRegistryProfile::Identity**

name	mult	type	description
identifier	1..1	URI	(NC) Globally unique identifier representing the persistent identity of the object. Note: The identifier is used to support lifecycle management, cross-system referencing, and comparison of objects across model exchanges and business processes. The identifier is expressed as a URI to allow compatibility with different identifier schemes. Profiles commonly restrict the identifier to a UUID to ensure global uniqueness and stable identity across distributed systems. Example: "urn:uuid:550e8400-e29b-41d4-a716-446655440000"

373

374 ### 3.12 (ConstrainedPrimitive) Language root class

375 Represents textual values that identifier a language based on a structured format that includes
376 language codes, optional region codes, and script subtags. This ensures consistency in
377 multilingual data processing, localization, metadata tagging, and content negotiation.
378 Example: Valid examples: "en", "fr", "es-ES", "zh-Hans", "nb-NO". Not valid examples:
379 "english", "Français", "norwegian", "zh-hant-cn" (invalid format).

380

381

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

Test data files are available in the CIM WG SharePoint.