IEC TS 61970-600-2

Edition 1.0 2017-07

TECHNICAL SPECIFICATION

colour inside

Energy management system application program interface (EMS-API) – Part 600-2: Common Grid Model Exchange Specification (CGMES) – Exchange profiles specification

IEC TS 61970-600-2:2017-07(en)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue
The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org
The world's leading online dictionary of electronic and electrical terms containing 20 000 terms and definitions in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary
65 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

IEC TS 61970-600-2

Edition 1.0 2017-07

TECHNICAL SPECIFICATION

colour inside

Energy management system application program interface (EMS-API) – Part 600-2: Common Grid Model Exchange Specification (CGMES) – Exchange profiles specification

INTERNATIONAL ELECTROTECHNICAL COMMISSION

ICS 33.200

ISBN 978-2-8322-4574-3

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

ENERGY MANAGEMENT SYSTEM APPLICATION PROGRAM INTERFACE (EMS-API) –

Part 600-2: Common Grid Model Exchange Specification (CGMES) – Exchange profiles specification

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

The main task of IEC technical committees is to prepare International Standards. In exceptional circumstances, a technical committee may propose the publication of a technical specification when

- the required support cannot be obtained for the publication of an International Standard, despite repeated efforts, or
- the subject is still under technical development or where, for any other reason, there is the future but no immediate possibility of an agreement on an International Standard.

Technical specifications are subject to review within three years of publication to decide whether they can be transformed into International Standards.

IEC TS 61970-600-2, which is a technical specification, has been prepared by IEC technical committee 57: Power systems management and associated information exchange.

The text of this technical specification is based on the following documents:

Enquiry draft	Report on voting
57/1816/DTS	57/1872/RVDTS

Full information on the voting for the approval of this technical specification can be found in the report on voting indicated in the above table.

This document has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all the parts in the IEC 61970 series, published under the general title *Energy Management System Application Program Interface (EMS-API)*, can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC website under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- transformed into an International standard,
- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

A bilingual version of this publication may be issued at a later date.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

ENERGY MANAGEMENT SYSTEM APPLICATION PROGRAM INTERFACE (EMS-API) –

Part 600-2: Common Grid Model Exchange Specification (CGMES) – Exchange profiles specification

1 Scope

This part of IEC 61970, which is a technical specification on the CGMES, details the requirements of the exchange profiles belonging to the CGMES. The related technical information and documentation (i.e. RDFS, OCL, XMI and HTML) needed for the implementation of the CGMES, which is not copyrighted by either IEC or CENELEC, is available at the ENTSO-E web site (www.entsoe.eu).

The CGMES is defined using information on the Common Information Model (CIM) available in the public domain.

The CGMES is a superset of the former ENTSO-E CIM based data exchange standard (Profile 1) which was based on CIM14 (UML14v02) and has been used for certain network models exchanges since 2009. The CGMES reflects TSO requirements (as known by 2014) for accurate modelling of the ENTSO-E area for power flow, short circuit and dynamics applications whilst also allowing for the exchange of any diagram layouts including GIS data of a grid model.

Future editions of this technical specification will be released to describe following CGMES versions which reflect the additional requirements due to European network codes or guidelines.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61968-4, *Application integration at electric utilities – System interfaces for distribution management – Part 4: Interfaces for records and asset management*

IEC 61970-301, *Energy management system application program interface (EMS-API) – Part 301: Common information model (CIM) base*

IEC 61970-302, *Energy management system application program interface (EMS-API) – Part 302: CIM for dynamics*[1]

IEC 61970-452, *Energy management system application program interface (EMS-API) – Part 452: CIM model exchange specification*

IEC 61970-453, *Energy management system application program interface (EMS-API) – Part 453: Diagram layout profile*

---

1 Under preparation. Stage at the time of publication: IEC/AFDIS 61970-302:2017.

IEC 61970-456, *Energy management system application program interface (EMS-API) – Part 456: Solved power system state profiles*

IEC 61970-552, *Energy management system application program interface (EMS-API) – Part 552: CIMXML Model exchange format*

3 Terms, definitions and abbreviated terms

For the purposes of this document, the following terms, definitions and abbreviated terms apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

NOTE For definitions which are not specified in the CGMES the definitions in the IEC CIM 61970 standards shall be applied.

3.1 Terms and definitions

3.1.1
Common Grid Model Exchange Specification
CGMES
specification used for the exchange of power system models between TSOs for the purpose of performing bilateral, regional or pan-European studies in the frame of TYNDP or TSOs' projects

Note 1 to entry: This is based on IEC CIM Standards and further extended to meet Network Codes' and projects' requirements. The standard defines a set of data model exchange profiles.

3.1.2
profile
uniquely named subset of classes, associations and attributes needed to accomplish a specific type of interface and based upon a canonical model

Note 1 to entry: The term profile may be used to define either the semantic model for an instance data payload or the syntactic schema for an instance data payload. A profile may be expressed in XSD, RDF, and/or OWL files. An instance data conforming to a profile can be tested in exchanges between applications. A profile is necessary in order to "use" the canonical model.

3.1.3
CIM Extension
collection of classes, attributes and associations, which extend the standard IEC CIM model in order to cover use cases not currently supported by IEC standards, and which are not considered to be international use cases or are covered by a later version of the standard which is not yet supported

3.1.4
ENTSO-E Extension
CIM Extension, specifically managed by ENTSO-E

3.1.5
boundary set
set containing all boundary points necessary for a given grid model exchange

Note 1 to entry: A Boundary set can have different coverage depending on the requirements of the common grid model exchange. A complete boundary set is necessary to assemble a pan-European power system model.

3.1.6
boundary point
BP
connection point between two Model Authority Sets (MAS)

Note 1 to entry: A Boundary point could be a ConnectivityNode or a TopologicalNode placed on a tie-line or in a substation. A Boundary point must be contained in a Boundary Set and must not be contained in the MAS of a TSO. A Boundary point is referenced by Terminals in the MAS of a TSO. ConnectivityNode and TopologicalNode are terms specified in IEC CIM standards. If a Boundary point is placed on a tie-line, the term X-Node is often used instead of Boundary point. X-Node is therefore a specific type of Boundary point.

3.2 Abbreviated terms

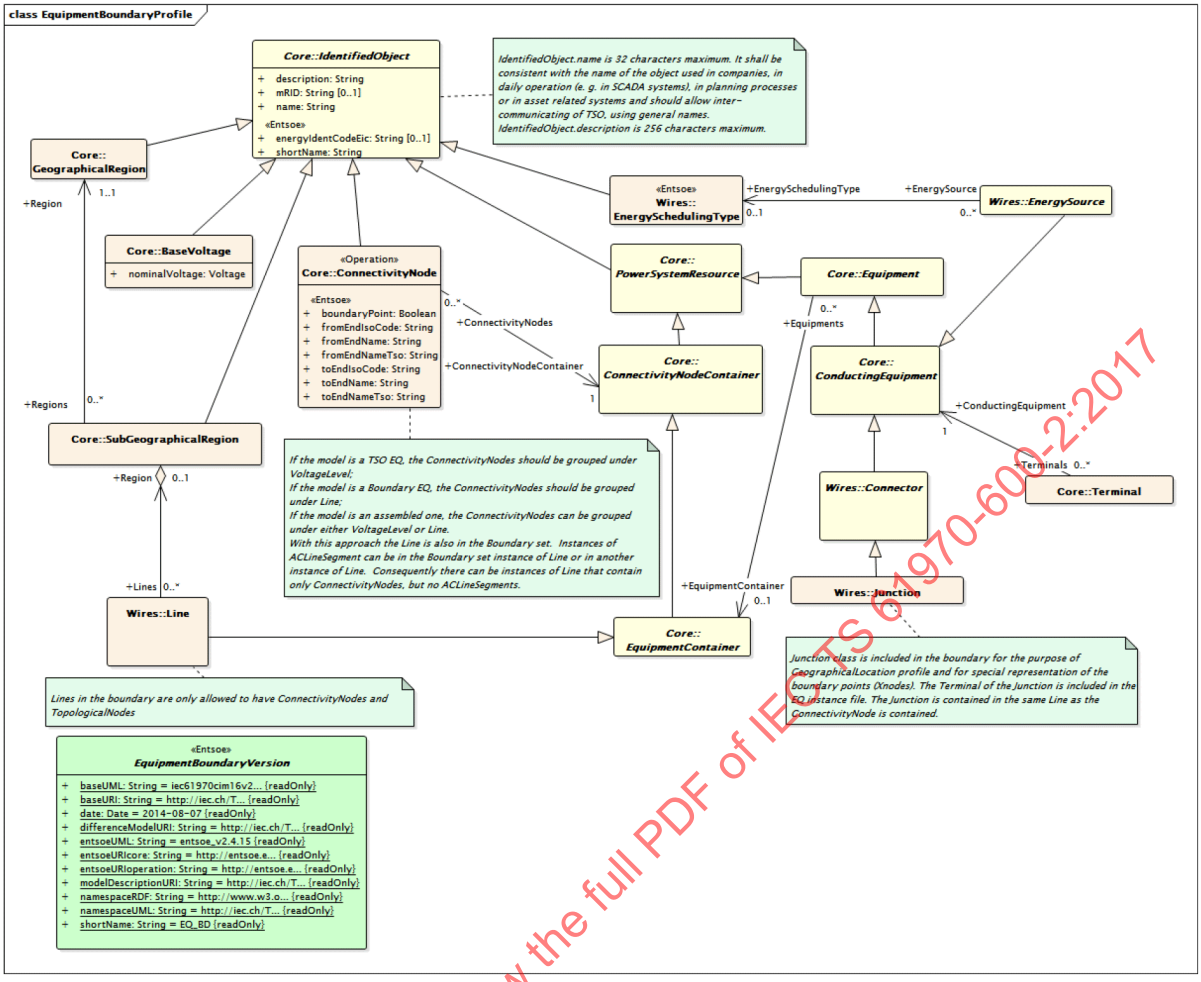
IEC	The International Electrotechnical Commission, headquartered in Geneva
TSO	Transmission System Operator
ENTSO-E	European Network of Transmission System Operators for Electricity (ENTSO-E has 43 TSO members)
MRID	CIM Master Resource Identifier
CIM	Common Information Model (electricity)
CGMES	Common Grid Model Exchange Specification
MAS	Model Authority Set
RDF	Resource Description Framework
EQ_BD	Boundary equipment profile or instance file
TP_BD	Boundary topology profile or instance file
EQ	Equipment profile or instance file
TP	Topology profile or instance file
SSH	Steady State Hypothesis profile or instance file
SV	State Variables profile or instance file
DL	Diagram Layout profile or instance file
GL	Geographical Location profile or instance file
BP	Boundary point

4 EquipmentBoundaryProfile

4.1 General

This profile has been built on the basis of IEC 61970-452 and adjusted to fit the purpose of the ENTSO-E boundary profile.

Figure 1 shows the diagram.



IEC

Figure 1 – diagram EquipmentBoundaryProfile

4.2 (Entsoe) EquipmentBoundaryVersion(Abstract)

Profile version details.

Table 1 shows all attributes of EquipmentBoundaryVersion.

Table 1 – Attributes of EquipmentBoundaryProfile::EquipmentBoundaryVersion

Attribute name	mult.	Attribute type	Description
baseUML (Entsoe)	1..1	String	ConstantValue= iec61970cim16v28_iec61968cim12v08_iec62325cim03v01a Base UML provided by CIM model manager.
baseURI (Entsoe)	1..1	String	ConstantValue= http://iec.ch/TC57/2013/61970-452/EquipmentBoundary/3 Profile URI used in the Model Exchange header and defined in IEC standards. It uniquely identifies the Profile and its version. It is given for information only and to identify the closest IEC profile to which this CGMES profile is based on.
date (Entsoe)	1..1	Date	ConstantValue= 2014-08-07 Profile creation date Form is YYYY-MM-DD for example for January 5, 2009 it is 2009-01-05.
differenceModelURI (Entsoe)	1..1	String	ConstantValue= http://iec.ch/TC57/61970-552/DifferenceModel/1# Difference model URI defined by IEC 61970-552.
entsoeUML (Entsoe)	1..1	String	ConstantValue= entsoe_v2.4.15 UML provided by ENTSO-E.
entsoeURIcore (Entsoe)	1..1	String	ConstantValue= http://entsoe.eu/CIM/EquipmentBoundary/3/1 Profile URI defined by ENTSO-E and used in the Model Exchange header. It uniquely identifies the Profile and its version. The last two elements in the URI (http://entsoe.eu/CIM/EquipmentBoundary/yy/zzz) indicate major and minor versions where: - yy – indicates a major version; - zzz – indicates a minor version.
entsoeURIoperation (Entsoe)	1..1	String	ConstantValue= http://entsoe.eu/CIM/EquipmentBoundaryOperation/3/1 Profile URI defined by ENTSO-E and used in the Model Exchange header. It uniquely identifies the Profile and its version. The last two elements in the URI (http://entsoe.eu/CIM/EquipmentBoundaryOperation/yy/zzz) indicate major and minor versions where: - yy – indicates a major version; - zzz – indicates a minor version.
modelDescriptionURI (Entsoe)	1..1	String	ConstantValue= http://iec.ch/TC57/61970-552/ModelDescription/1# Model Description URI defined by IEC 61970-552.
namespaceRDF (Entsoe)	1..1	String	ConstantValue= http://www.w3.org/1999/02/22-rdf-syntax-ns# RDF namespace.
namespaceUML (Entsoe)	1..1	String	ConstantValue= http://iec.ch/TC57/2013/CIM-schema-cim16# CIM UML namespace.
shortName (Entsoe)	1..1	String	ConstantValue= EQ_BD The short name of the profile used in profile documentation.

4.3 Core

4.3.1 ConductingEquipment(Abstract)

The parts of the AC power system that are designed to carry current or that are conductively connected through terminals.

Inheritance path: ->Equipment->PowerSystemResource->IdentifiedObject

Table 2 shows all attributes of ConductingEquipment.

Table 2 – Attributes of Core::ConductingEquipment

Attribute name	mult.	Attribute type	Description
description	1..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	1..1	String	see IdentifiedObject

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

Table 3 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

4.3.2 (Operation) ConnectivityNode

Connectivity nodes are points where terminals of AC conducting equipment are connected together with zero impedance.

If the model is a TSO EQ, the ConnectivityNodes should be grouped under VoltageLevel; If the model is a Boundary EQ, the ConnectivityNodes should be grouped under Line; If the model is an assembled one, the ConnectivityNodes can be grouped under either VoltageLevel or Line. With this approach the Line is also in the Boundary set. Instances of ACLineSegment can be in the Boundary set instance of Line or in another instance of Line. Consequently there can be instances of Line that contain only ConnectivityNodes, but no ACLineSegments.

Inheritance path: ->IdentifiedObject

Table 4 shows all attributes of ConnectivityNode.

Table 4 – Attributes of Core::ConnectivityNode

Attribute name	mult.	Attribute type	Description
boundaryPoint (Entsoe)	1..1	Boolean	Identifies if a node is a BoundaryPoint. If boundaryPoint=true the ConnectivityNode or the TopologicalNode represents a BoundaryPoint.
fromEndIsoCode (Entsoe)	1..1	String	The attribute is used for an exchange of the ISO code of the region to which the “From” side of the Boundary point belongs to or it is connected to. The ISO code is two characters country code as defined by ISO 3166 (http://www.iso.org/iso/country_codes). The length of the string is 2 characters maximum. The attribute is a required for the Boundary Model Authority Set where this attribute is used only for the TopologicalNode in the Boundary Topology profile and ConnectivityNode in the Boundary Equipment profile.
fromEndName (Entsoe)	1..1	String	The attribute is used for an exchange of a human readable name with length of the string 32 characters maximum. The attribute covers two cases: if the Boundary point is placed on a tie-line the attribute is used for exchange of the geographical name of the substation to which the “From” side of the tie-line is connected to. if the Boundary point is placed in a substation the attribute is used for exchange of the name of the element (e.g. PowerTransformer, ACLineSegment, Switch, etc) to which the “From” side of the Boundary point is connected to. The attribute is required for the Boundary Model Authority Set where it is used only for the TopologicalNode in the Boundary Topology profile and ConnectivityNode in the Boundary Equipment profile.
fromEndNameTso (Entsoe)	1..1	String	The attribute is used for an exchange of the name of the TSO to which the “From” side of the Boundary point belongs to or it is connected to. The length of the string is 32 characters maximum. The attribute is required for the Boundary Model Authority Set where it is used only for the TopologicalNode in the Boundary Topology profile and ConnectivityNode in the Boundary Equipment profile.
toEndIsoCode (Entsoe)	1..1	String	The attribute is used for an exchange of the ISO code of the region to which the “To” side of the Boundary point belongs to or it is connected to. The ISO code is two characters country code as defined by ISO 3166 (http://www.iso.org/iso/country_codes). The length of the string is 2 characters maximum. The attribute is a required for the Boundary Model Authority Set where this attribute is used only for the TopologicalNode in the Boundary Topology profile and ConnectivityNode in the Boundary Equipment profile.
toEndName (Entsoe)	1..1	String	The attribute is used for an exchange of a human readable name with length of the string 32 characters maximum. The attribute covers two cases: if the Boundary point is placed on a tie-line the attribute is used for exchange of the geographical name of the substation to which the “To” side of the tie-line is connected to. if the Boundary point is placed in a substation the attribute is used for exchange of the name of the **element** (e.g. PowerTransformer, ACLineSegment, Switch, etc) to which the “To” side of the Boundary point is connected to. The attribute is required for the Boundary Model Authority Set where it is used only for the TopologicalNode in the Boundary Topology profile and ConnectivityNode in the Boundary Equipment profile.

Attribute name	mult.	Attribute type	Description
toEndNameTso (Entsoe)	1..1	String	The attribute is used for an exchange of the name of the TSO to which the “To” side of the Boundary point belongs to or it is connected to. The length of the string is 32 characters maximum. The attribute is required for the Boundary Model Authority Set where it is used only for the TopologicalNode in the Boundary Topology profile and ConnectivityNode in the Boundary Equipment profile.
description	1..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	1..1	String	see IdentifiedObject

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

Table 5 – Association ends of Core::ConnectivityNode with other classes

mult.	mult.	Class type name	Description
ConnectivityNodeContainer (Operation)	[1..1]	ConnectivityNodeContainer	Container of this connectivity node. Association Based On: Core:: ConnectivityNode.ConnectivityNodes[0..*] ----- Core:: ConnectivityNodeContainer.ConnectivityNodeContainer[1]

4.3.3 ConnectivityNodeContainer(Abstract)

A base class for all objects that may contain connectivity nodes or topological nodes.

Inheritance path: ->PowerSystemResource->IdentifiedObject

Table 6 shows all attributes of ConnectivityNodeContainer.

Table 6 – Attributes of Core::ConnectivityNodeContainer

Attribute name	mult.	Attribute type	Description
description	1..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	1..1	String	see IdentifiedObject

4.3.4 Equipment(Abstract)

The parts of a power system that are physical devices, electronic or mechanical.

Inheritance path: ->PowerSystemResource->IdentifiedObject

Table 7 shows all attributes of Equipment.

Table 7 – Attributes of Core::Equipment

Attribute name	mult.	Attribute type	Description
description	1..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	1..1	String	see IdentifiedObject

Table 8 shows all association ends of Equipment with other classes.

Table 8 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	[0..1]	EquipmentContainer	Container of this equipment. Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

4.3.5 EquipmentContainer(Abstract)

A modeling construct to provide a root class for containing equipment.

Inheritance path: ->ConnectivityNodeContainer->PowerSystemResource->IdentifiedObject

Table 9 shows all attributes of EquipmentContainer.

Table 9 – Attributes of Core::EquipmentContainer

Attribute name	mult.	Attribute type	Description
description	1..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	1..1	String	see IdentifiedObject

4.3.6 PowerSystemResource(Abstract)

A power system resource can be an item of equipment such as a switch, an equipment container containing many individual items of equipment such as a substation, or an organisational entity such as sub-control area. Power system resources can have measurements associated.

Inheritance path: ->IdentifiedObject

Table 10 shows all attributes of PowerSystemResource.

Table 10 – Attributes of Core::PowerSystemResource

Attribute name	mult.	Attribute type	Description
description	1..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	1..1	String	see IdentifiedObject

4.3.7 Terminal

An AC electrical connection point to a piece of conducting equipment. Terminals are connected at physical connection points called connectivity nodes.

4.3.8 BaseVoltage

Defines a system base voltage which is referenced.

Inheritance path: ->IdentifiedObject

Table 11 shows all attributes of BaseVoltage.

Table 11 – Attributes of Core::BaseVoltage

Attribute name	mult.	Attribute type	Description
nominalVoltage	1..1	Voltage	The power system resource's base voltage.
description	1..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	1..1	String	see IdentifiedObject

4.3.9 GeographicalRegion

A geographical region of a power system network model.

Inheritance path: ->IdentifiedObject

Table 12 shows all attributes of GeographicalRegion.

Table 12 – Attributes of Core::GeographicalRegion

Attribute name	mult.	Attribute type	Description
description	1..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	1..1	String	see IdentifiedObject

4.3.10 IdentifiedObject(Abstract)

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

IdentifiedObject.name is 32 characters maximum. It shall be consistent with the name of the object used in companies, in daily operation (e. g. in SCADA systems), in planning processes or in asset related systems and should allow inter-communicating of TSO, using general names. IdentifiedObject.description is 256 characters maximum.

Table 13 shows all attributes of IdentifiedObject.

Table 13 – Attributes of Core::IdentifiedObject

Attribute name	mult.	Attribute type	Description
description	1..1	String	The description is a free human readable text describing or naming the object. It may be non unique and may not correlate to a naming hierarchy.
energyIdentCodeEic (Entsoe)	0..1	String	The attribute is used for an exchange of the EIC code (Energy identification Code). The length of the string is 16 characters as defined by the EIC code. References: Local issuing offices for EIC: https://www.entsoe.eu/publications/edi-library/links-to-eic-websites/ EIC description: https://www.entsoe.eu/index.php?id=73&libCat=eic .
mRID	0..1	String	Master resource identifier issued by a model authority. The mRID is globally 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 61970-552:2013, the mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements.
name	1..1	String	The name is any free human readable and possibly non unique text naming the object.
shortName (Entsoe)	1..1	String	The attribute is used for an exchange of a human readable short name with length of the string 12 characters maximum.

4.3.11 SubGeographicalRegion

A subset of a geographical region of a power system network model.

Inheritance path: ->IdentifiedObject

Table 14 shows all attributes of SubGeographicalRegion.

Table 14 – Attributes of Core::SubGeographicalRegion

Attribute name	mult.	Attribute type	Description
description	1..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	1..1	String	see IdentifiedObject

Table 15 shows all association ends of SubGeographicalRegion with other classes.

Table 15 – Association ends of Core::SubGeographicalRegion with other classes

mult.	mult.	Class type name	Description
Region	[1..1]	GeographicalRegion	The geographical region to which this sub-geographical region is within. Association Based On: Core:: GeographicalRegion.Region[0..1] ----- Core:: SubGeographicalRegion.Regions[0..*]

4.4 Wires

4.4.1 Connector(Abstract)

A conductor, or group of conductors, with negligible impedance, that serve to connect other conducting equipment within a single substation and are modelled with a single logical terminal.

Inheritance path: ->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 16 shows all attributes of Connector.

Table 16 – Attributes of Wires::Connector

Attribute name	mult.	Attribute type	Description
description	1..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	1..1	String	see IdentifiedObject

Table 17 shows all association ends of Equipment with other classes.

Table 17 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] --- Core:: Equipment.Equipments[0..*]

4.4.2 (Entsoe) EnergySchedulingType

Used to define the type of generation for scheduling purposes.

Inheritance path: ->IdentifiedObject

Table 18 shows all attributes of EnergySchedulingType.

Table 18 – Attributes of Wires::EnergySchedulingType

Attribute name	mult.	Attribute type	Description
description	1..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	1..1	String	see IdentifiedObject

4.4.3 EnergySource(Abstract)

A generic equivalent for an energy supplier on a transmission or distribution voltage level.

Inheritance path: ->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 19 shows all attributes of EnergySource.

Table 19 – Attributes of Wires::EnergySource

Attribute name	mult.	Attribute type	Description
description	1..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	1..1	String	see IdentifiedObject

Table 20 shows all association ends of EnergySource with other classes.

Table 20 – Association ends of Wires::EnergySource with other classes

mult.	mult.	Class type name	Description
EnergySchedulingType	[0..1]	EnergySchedulingType	Energy Scheduling Type of an Energy Source Association Based On: Wires:: EnergySource.EnergySource[0..*] ----- Extension:: EnergySchedulingType.EnergySchedulingType[0..1]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

4.4.4 Junction

A point where one or more conducting equipments are connected with zero resistance.

Junction class is included in the boundary for the purpose of GeographicalLocation profile and for special representation of the boundary points (Xnodes). The Terminal of the Junction is included in the EQ instance file. The Junction is contained in the same Line as the ConnectivityNode is contained.

Inheritance path: ->Connector->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 21 shows all attributes of Junction.

Table 21 – Attributes of Wires::Junction

Attribute name	mult.	Attribute type	Description
description	1..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	1..1	String	see IdentifiedObject

Table 22 shows all association ends of Equipment with other classes.

Table 22 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

4.4.5 Line

Contains equipment beyond a substation belonging to a power transmission line.

Lines in the boundary are only allowed to have ConnectivityNodes and TopologicalNodes

Inheritance path: ->EquipmentContainer->ConnectivityNodeContainer->PowerSystemResource->IdentifiedObject

Table 23 shows all attributes of Line.

Table 23 – Attributes of Wires::Line

Attribute name	mult.	Attribute type	Description
description	1..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	1..1	String	see IdentifiedObject

Table 24 shows all association ends of Line with other classes.

Table 24 – Association ends of Wires::Line with other classes

mult.	mult.	Class type name	Description
Region	[0..1]	SubGeographicalRegion	The lines within the sub-geographical region. Association Based On: Core:: SubGeographicalRegion.Region[0..1] ----- Wires:: Line.Lines[0..*]

4.5 Global Used Datatypes

4.5.1 Datatypes – Voltage

Electrical voltage, can be both AC and DC.

Value type is IEEE 754 simple precision floating point

Table 25 shows all attributes of Voltage.

Table 25 – Attributes of DomainProfile::Voltage.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= V
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= k

4.5.2 Enumerations

4.5.2.1 UnitSymbol

The units defined for usage in the CIM.

Literal	Description	Code
VA	Apparent power in volt ampere.	
W	Active power in watt.	
VAr	Reactive power in volt ampere reactive.	
VAh	Apparent energy in volt ampere hours.	
Wh	Real energy in what hours.	
VArh	Reactive energy in volt ampere reactive hours.	
V	Voltage in volt.	
ohm	Resistance in ohm.	
A	Current in ampere.	
F	Capacitance in farad.	
H	Inductance in henry.	
degC	Relative temperature in degrees Celsius. In the SI unit system the symbol is ºC. Electric charge is measured in coulomb that has the unit symbol C. To distinguish degree Celsius form coulomb the symbol used in the UML is degC. Reason for not using ºC is the special character º is difficult to manage in software.	
s	Time in seconds.	
min	Time in minutes.	
h	Time in hours.	
deg	Plane angle in degrees.	
rad	Plane angle in radians.	
J	Energy in joule.	
N	Force in newton.	
S	Conductance in siemens.	
none	Dimension less quantity, e.g. count, per unit, etc.	
Hz	Frequency in hertz.	
g	Mass in gram.	
Pa	Pressure in pascal (n/m2).	
m	Length in meter.	
m2	Area in square meters.	
m3	Volume in cubic meters.	

4.5.2.2 UnitMultiplier

The unit multipliers defined for the CIM.

Literal	Description	Code
p	Pico 10**-12.	
n	Nano 10**-9.	
micro	Micro 10**-6.	
m	Milli 10**-3.	
c	Centi 10**-2.	
d	Deci 10**-1.	
k	Kilo 10**3.	
M	Mega 10**6.	
G	Giga 10**9.	
T	Tera 10**12.	
none	No multiplier or equivalently multiply by 1.	

4.5.3 Primitives

4.5.3.1 String

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

4.5.3.2 Date

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".

4.5.3.3 Boolean

A type with the value space "true" and "false".

4.5.3.4 Float

A floating point number. The range is unspecified and not limited.

5 TopologyBoundaryProfile

5.1 General

This profile has been built on the basis of the IEC 61970-456 document and adjusted to fit the purpose of the ENTSO-E boundary profile.

Figure 2 shows the diagram.

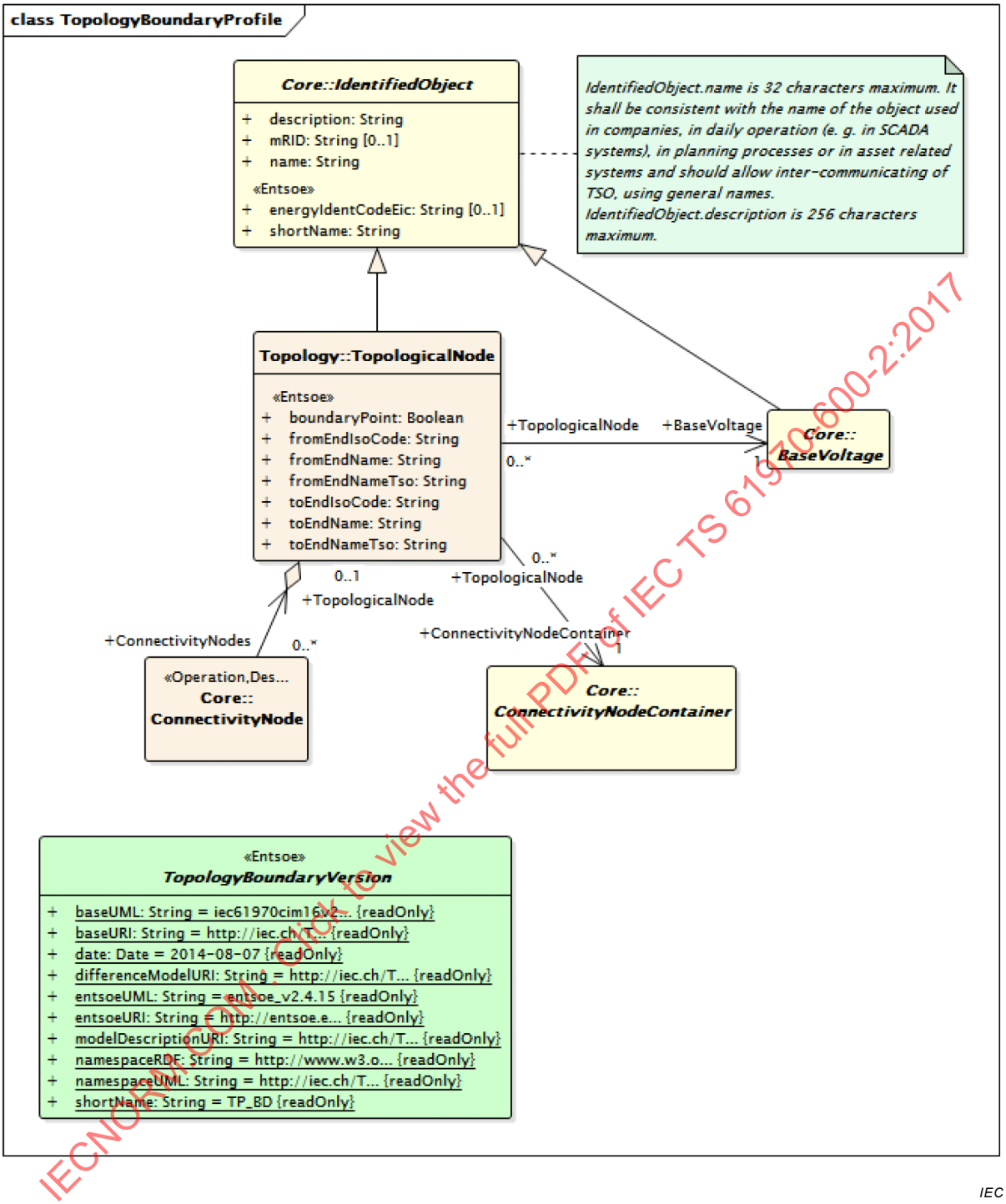


Figure 2 – diagram TopologyBoundaryProfile

5.2 (Entsoe) TopologyBoundaryVersion(Abstract)

Version details.

Table 26 shows all attributes of TopologyBoundaryVersion.

Table 26 – Attributes of TopologyBoundaryProfile::TopologyBoundaryVersion

Attribute name	mult.	Attribute type	Description
baseUML (Entsoe)	1..1	String	ConstantValue= iec61970cim16v28_iec61968cim12v08_iec62325cim03v01a Base UML provided by CIM model manager.
baseURI (Entsoe)	1..1	String	ConstantValue= http://iec.ch/TC57/2013/61970-456/TopologyBoundary/3 Profile URI used in the Model Exchange header and defined in IEC standards. It uniquely identifies the Profile and its version. It is given for information only and to identify the closest IEC profile to which this CGMES profile is based on.
date (Entsoe)	1..1	Date	ConstantValue= 2014-08-07 Profile creation date Form is YYYY-MM-DD for example for January 5, 2009 it is 2009-01-05.
differenceModelURI (Entsoe)	1..1	String	ConstantValue= http://iec.ch/TC57/61970-552/DifferenceModel/1# Difference model URI defined by IEC 61970-552.
entsoeUML (Entsoe)	1..1	String	ConstantValue= entsoe_v2.4.15 UML provided by ENTSO-E.
entsoeURI (Entsoe)	1..1	String	ConstantValue= http://entsoe.eu/CIM/TopologyBoundary/3/1 Profile URI defined by ENTSO-E and used in the Model Exchange header. It uniquely identifies the Profile and its version. The last two elements in the URI (http://entsoe.eu/CIM/TopologyBoundary/yy/zzz) indicate major and minor versions where: - yy – indicates a major version; - zzz – indicates a minor version.
modelDescriptionURI (Entsoe)	1..1	String	ConstantValue= http://iec.ch/TC57/61970-552/ModelDescription/1# Model Description URI defined by IEC 61970-552.
namespaceRDF (Entsoe)	1..1	String	ConstantValue= http://www.w3.org/1999/02/22-rdf-syntax-ns# RDF namespace.
namespaceUML (Entsoe)	1..1	String	ConstantValue= http://iec.ch/TC57/2013/CIM-schema-cim16# CIM UML namespace.
shortName (Entsoe)	1..1	String	ConstantValue= TP_BD The short name of the profile used in profile documentation.

5.3 Core

5.3.1 BaseVoltage(Abstract)

Defines a system base voltage which is referenced.

Inheritance path: ->IdentifiedObject

Table 27 shows all attributes of BaseVoltage.

Table 27 – Attributes of Core::BaseVoltage

Attribute name	mult.	Attribute type	Description
description	1..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	1..1	String	see IdentifiedObject

5.3.2 (Operation,Description) ConnectivityNode

Connectivity nodes are points where terminals of AC conducting equipment are connected together with zero impedance.

OCL constraint: R.4.10.11. Description length restriction

OCL constraint: R.4.10.11. Energy Ident Code length restriction (optional)

OCL constraint: R.4.10.11. From End ISO Code length restriction

OCL constraint: R.4.10.11. From End Name length restriction

OCL constraint: R.4.10.11. From End Name TSO length restriction

OCL constraint: R.4.10.11. Name length restriction

OCL constraint: R.4.10.11. ShortName length restriction

OCL constraint: R.4.10.11. To End ISO Code length restriction

OCL constraint: R.4.10.11. To End Name length restriction

OCL constraint: R.4.10.11. To End Name TSO length restriction

5.3.3 ConnectivityNodeContainer(Abstract)

A base class for all objects that may contain connectivity nodes or topological nodes.

5.3.4 IdentifiedObject(Abstract)

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

IdentifiedObject.name is 32 characters maximum. It shall be consistent with the name of the object used in companies, in daily operation (e. g. in SCADA systems), in planning processes or in asset related systems and should allow inter-communicating of TSO, using general names. IdentifiedObject.description is 256 characters maximum.

Table 28 shows all attributes of IdentifiedObject.

Table 28 – Attributes of Core::IdentifiedObject

Attribute name	mult.	Attribute type	Description
description	1..1	String	The description is a free human readable text describing or naming the object. It may be non unique and may not correlate to a naming hierarchy.
energyIdentCodeEic (Entsoe)	0..1	String	The attribute is used for an exchange of the EIC code (Energy identification Code). The length of the string is 16 characters as defined by the EIC code. References: Local issuing offices for EIC: https://www.entsoe.eu/publications/edi-library/links/links-to-eic-websites/ EIC description: https://www.entsoe.eu/index.php?id=73&libCat=eic .
mRID	0..1	String	Master resource identifier issued by a model authority. The mRID is globally 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 61970-552:2013, the mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements.
name	1..1	String	The name is any free human readable and possibly non unique text naming the object.
shortName (Entsoe)	1..1	String	The attribute is used for an exchange of a human readable short name with length of the string 12 characters maximum.

5.4 Topology – TopologicalNode

For a detailed substation model a topological node is a set of connectivity nodes that, in the current network state, are connected together through any type of closed switches, including jumpers. Topological nodes change as the current network state changes (i.e., switches, breakers, etc. change state). For a planning model, switch statuses are not used to form topological nodes. Instead they are manually created or deleted in a model builder tool. Topological nodes maintained this way are also called "busses".

OCL constraint: R.4.10.11. Description length restriction

OCL constraint: R.4.10.11. Energy Ident Code length restriction (optional)

OCL constraint: R.4.10.11. From End ISO Code length restriction

OCL constraint: R.4.10.11. From End Name length restriction

OCL constraint: R.4.10.11. From End Name TSO length restriction

OCL constraint: R.4.10.11. Name length restriction

OCL constraint: R.4.10.11. ShortName length restriction

OCL constraint: R.4.10.11. To End ISO Code length restriction

OCL constraint: R.4.10.11. To End Name length restriction

OCL constraint: R.4.10.11. To End Name TSO length restriction

Inheritance path: ->IdentifiedObject

Table 29 shows all attributes of TopologicalNode.

Table 29 – Attributes of Topology::TopologicalNode

Attribute name	mult.	Attribute type	Description
boundaryPoint (Entsoe)	1..1	Boolean	Identifies if a node is a BoundaryPoint. If boundaryPoint=true the ConnectivityNode or the TopologicalNode represents a BoundaryPoint.
fromEndIsoCode (Entsoe)	1..1	String	The attribute is used for an exchange of the ISO code of the region to which the “From” side of the Boundary point belongs to or it is connected to. The ISO code is two characters country code as defined by ISO 3166 (http://www.iso.org/iso/country_codes). The length of the string is 2 characters maximum. The attribute is a required for the Boundary Model Authority Set where this attribute is used only for the TopologicalNode in the Boundary Topology profile and ConnectivityNode in the Boundary Equipment profile.
fromEndName (Entsoe)	1..1	String	The attribute is used for an exchange of a human readable name with length of the string 32 characters maximum. The attribute covers two cases: if the Boundary point is placed on a tie-line the attribute is used for exchange of the geographical name of the substation to which the “From” side of the tie-line is connected to. if the Boundary point is placed in a substation the attribute is used for exchange of the name of the **element** (e.g. PowerTransformer, ACLineSegment, Switch, etc) to which the “From” side of the Boundary point is connected to. The attribute is required for the Boundary Model Authority Set where it is used only for the TopologicalNode in the Boundary Topology profile and ConnectivityNode in the Boundary Equipment profile.
fromEndNameTso (Entsoe)	1..1	String	The attribute is used for an exchange of the name of the TSO to which the “From” side of the Boundary point belongs to or it is connected to. The length of the string is 32 characters maximum. The attribute is required for the Boundary Model Authority Set where it is used only for the TopologicalNode in the Boundary Topology profile and ConnectivityNode in the Boundary Equipment profile.
toEndIsoCode (Entsoe)	1..1	String	The attribute is used for an exchange of the ISO code of the region to which the “To” side of the Boundary point belongs to or it is connected to. The ISO code is two characters country code as defined by ISO 3166 (http://www.iso.org/iso/country_codes). The length of the string is 2 characters maximum. The attribute is a required for the Boundary Model Authority Set where this attribute is used only for the TopologicalNode in the Boundary Topology profile and ConnectivityNode in the Boundary Equipment profile.
toEndName (Entsoe)	1..1	String	The attribute is used for an exchange of a human readable name with length of the string 32 characters maximum. The attribute covers two cases: if the Boundary point is placed on a tie-line the attribute is used for exchange of the geographical name of the substation to which the “To” side of the tie-line is connected to. if the Boundary point is placed in a substation the attribute is used for exchange of the name of the element (e.g. PowerTransformer, ACLineSegment, Switch, etc) to which the “To” side of the Boundary point is connected to. The attribute is required for the Boundary Model Authority Set where it is used only for the TopologicalNode in the Boundary Topology profile and ConnectivityNode in the Boundary Equipment profile.

Attribute name	mult.	Attribute type	Description
toEndNameTso (Entsoe)	1..1	String	The attribute is used for an exchange of the name of the TSO to which the “To” side of the Boundary point belongs to or it is connected to. The length of the string is 32 characters maximum. The attribute is required for the Boundary Model Authority Set where it is used only for the TopologicalNode in the Boundary Topology profile and ConnectivityNode in the Boundary Equipment profile.
description	1..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	1..1	String	see IdentifiedObject

Table 30 shows all association ends of TopologicalNode with other classes.

Table 30 – Association ends of Topology::TopologicalNode with other classes

mult.	mult.	Class type name	Description
ConnectivityNodeContainer	[1..1]	ConnectivityNodeContainer	The connectivity node container to which the toplogical node belongs. Association Based On: Topology:: TopologicalNode.TopologicalNode[0..*] ----- Core:: ConnectivityNodeContainer.ConnectivityNodeContainer[0..1]
BaseVoltage	[1..1]	BaseVoltage	The base voltage of the topologocial node. Association Based On: Topology:: TopologicalNode.TopologicalNode[0..*] ----- Core:: BaseVoltage.BaseVoltage[0..1]

5.5 Global Used Datatypes – Primitives

5.5.1 String

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

5.5.2 Date

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".

5.5.3 Boolean

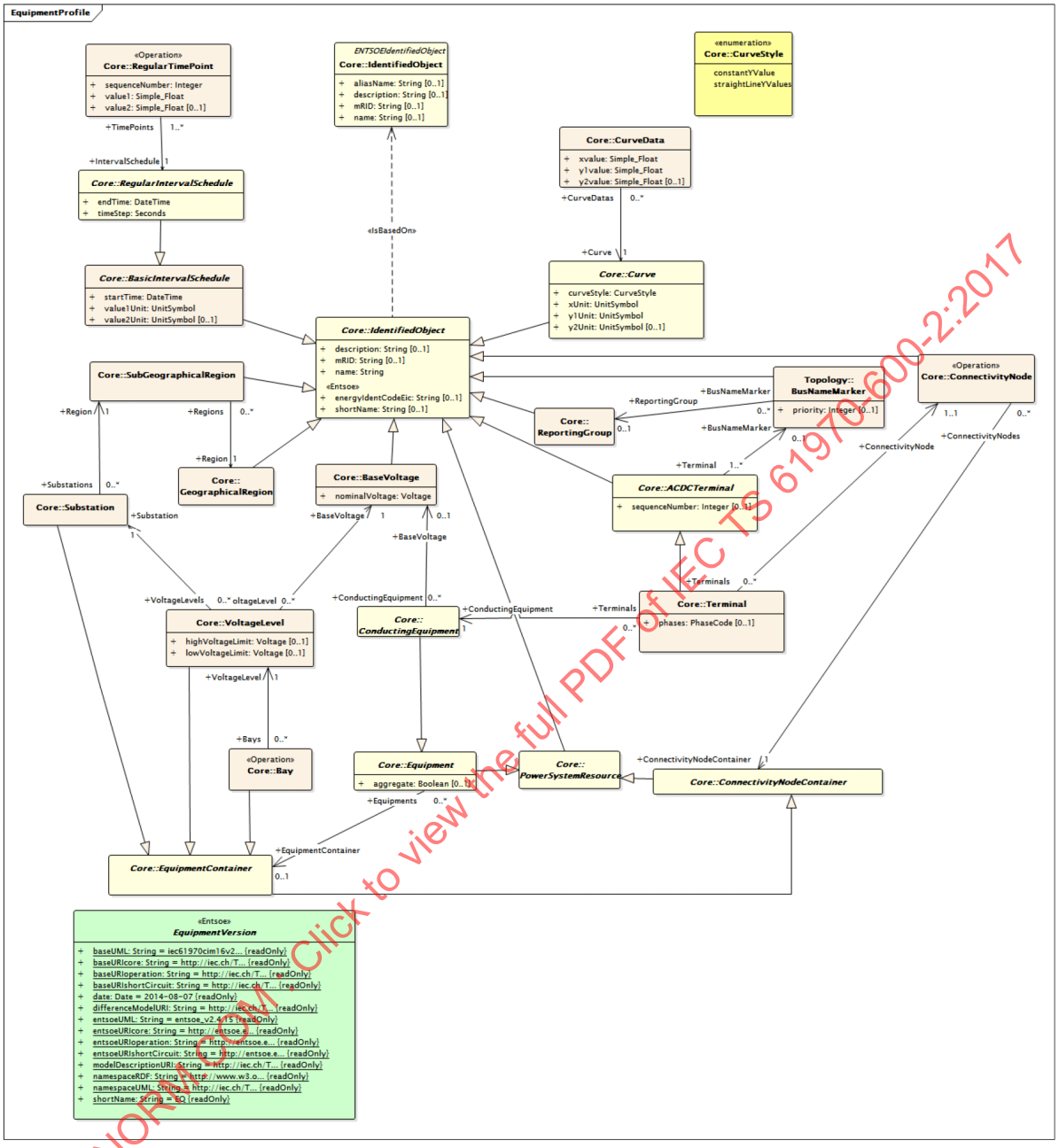
A type with the value space "true" and "false".

6 EquipmentProfile

6.1 General

This profile has been built on the basis of the IEC 61970-452 document and adjusted to fit the purpose of the ENTSO-E CGMES.

Figure 3 shows the diagram.



IEC

Figure 3 – diagram EquipmentProfile

6.2 (Entsoe) EquipmentVersion(Abstract)

Version details.

Table 31 shows all attributes of EquipmentVersion.

Table 31 – Attributes of EquipmentProfile::EquipmentVersion

Attribute name	mult.	Attribute type	Description
baseUML (Entsoe)	1..1	String	ConstantValue= iec61970cim16v28_iec61968cim12v08_iec62325cim03v01a Base UML provided by CIM model manager.
baseURIcore (Entsoe)	1..1	String	ConstantValue= http://iec.ch/TC57/2013/61970-452/EquipmentCore/4 Profile URI used in the Model Exchange header and defined in IEC standards. It uniquely identifies the Profile and its version. It is given for information only and to identify the closest IEC profile to which this CGMES profile is based on.
baseURIoperation (Entsoe)	1..1	String	ConstantValue= http://iec.ch/TC57/2013/61970-452/EquipmentOperation/4 Profile URI used in the Model Exchange header and defined in IEC standards. It uniquely identifies the Profile and its version. It is given for information only and to identify the closest IEC profile to which this CGMES profile is based on.
baseURIshortCircuit (Entsoe)	1..1	String	ConstantValue= http://iec.ch/TC57/2013/61970-452/EquipmentShortCircuit/4 Profile URI used in the Model Exchange header and defined in IEC standards. It uniquely identifies the Profile and its version. It is given for information only and to identify the closest IEC profile to which this CGMES profile is based on.
date (Entsoe)	1..1	Date	ConstantValue= 2014-08-07 Profile creation date Form is YYYY-MM-DD for example for January 5, 2009 it is 2009-01-05.
differenceModelURI (Entsoe)	1..1	String	ConstantValue= http://iec.ch/TC57/61970-552/DifferenceModel/1# Difference model URI defined by IEC 61970-552.
entsoeUML (Entsoe)	1..1	String	ConstantValue= entsoe_v2.4.15 UML provided by ENTSO-E.
entsoeURIcore (Entsoe)	1..1	String	ConstantValue= http://entsoe.eu/CIM/EquipmentCore/3/1 Profile URI defined by ENTSO-E and used in the Model Exchange header. It uniquely identifies the Profile and its version. The last two elements in the URI (http://entsoe.eu/CIM/EquipmentCore/yy/zzz) indicate major and minor versions where: - yy – indicates a major version; - zzz – indicates a minor version.
entsoeURIoperation (Entsoe)	1..1	String	ConstantValue= http://entsoe.eu/CIM/EquipmentOperation/3/1 Profile URI defined by ENTSO-E and used in the Model Exchange header. It uniquely identifies the Profile and its version. The last two elements in the URI (http://entsoe.eu/CIM/EquipmentOperation/yy/zzz) indicate major and minor versions where: - yy – indicates a major version; - zzz – indicates a minor version.
entsoeURIshortCircuit (Entsoe)	1..1	String	ConstantValue= http://entsoe.eu/CIM/EquipmentShortCircuit/3/1 Profile URI defined by ENTSO-E and used in the Model Exchange header. It uniquely identifies the Profile and its version. The last two elements in the URI (http://entsoe.eu/CIM/EquipmentShortCircuit/yy/zzz) indicate major and minor versions where: - yy – indicates a major version; - zzz – indicates a minor version.
modelDescriptionURI (Entsoe)	1..1	String	ConstantValue= http://iec.ch/TC57/61970-552/ModelDescription/1# Model Description URI defined by IEC 61970-552.

Attribute name	mult.	Attribute type	Description
namespaceRDF (Entsoe)	1..1	String	ConstantValue= http://www.w3.org/1999/02/22-rdf-syntax-ns# RDF namespace.
namespaceUML (Entsoe)	1..1	String	ConstantValue= http://iec.ch/TC57/2013/CIM-schema-cim16# CIM UML namespace.
shortName (Entsoe)	1..1	String	ConstantValue= EQ The short name of the profile used in profile documentation.

6.3 DC

6.3.1 General

Figure 4 shows the diagram.

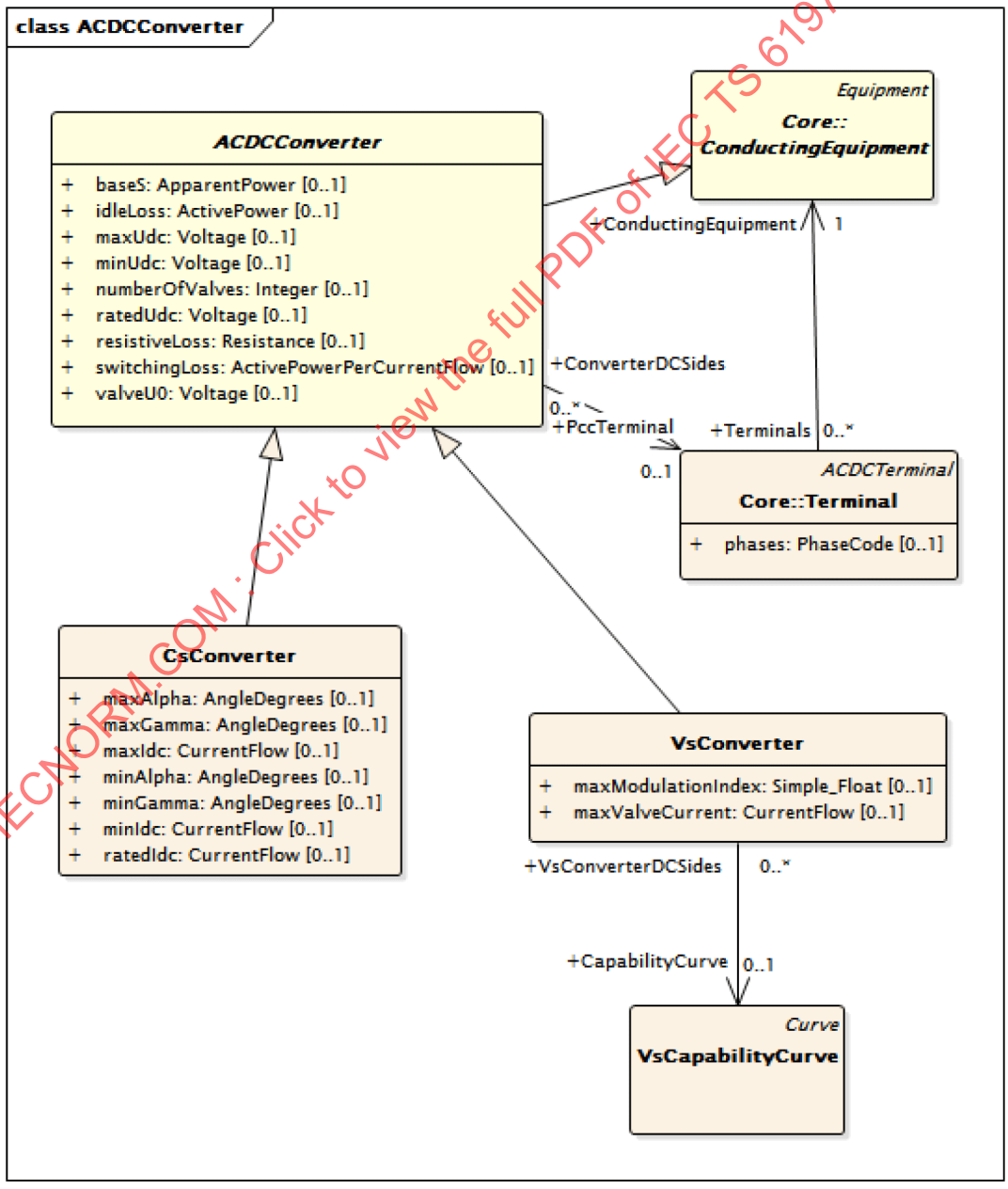


Figure 4 – diagram ACDCConverter

This diagram shows classes used to model DC equipment.

Figure 5 shows the diagram.

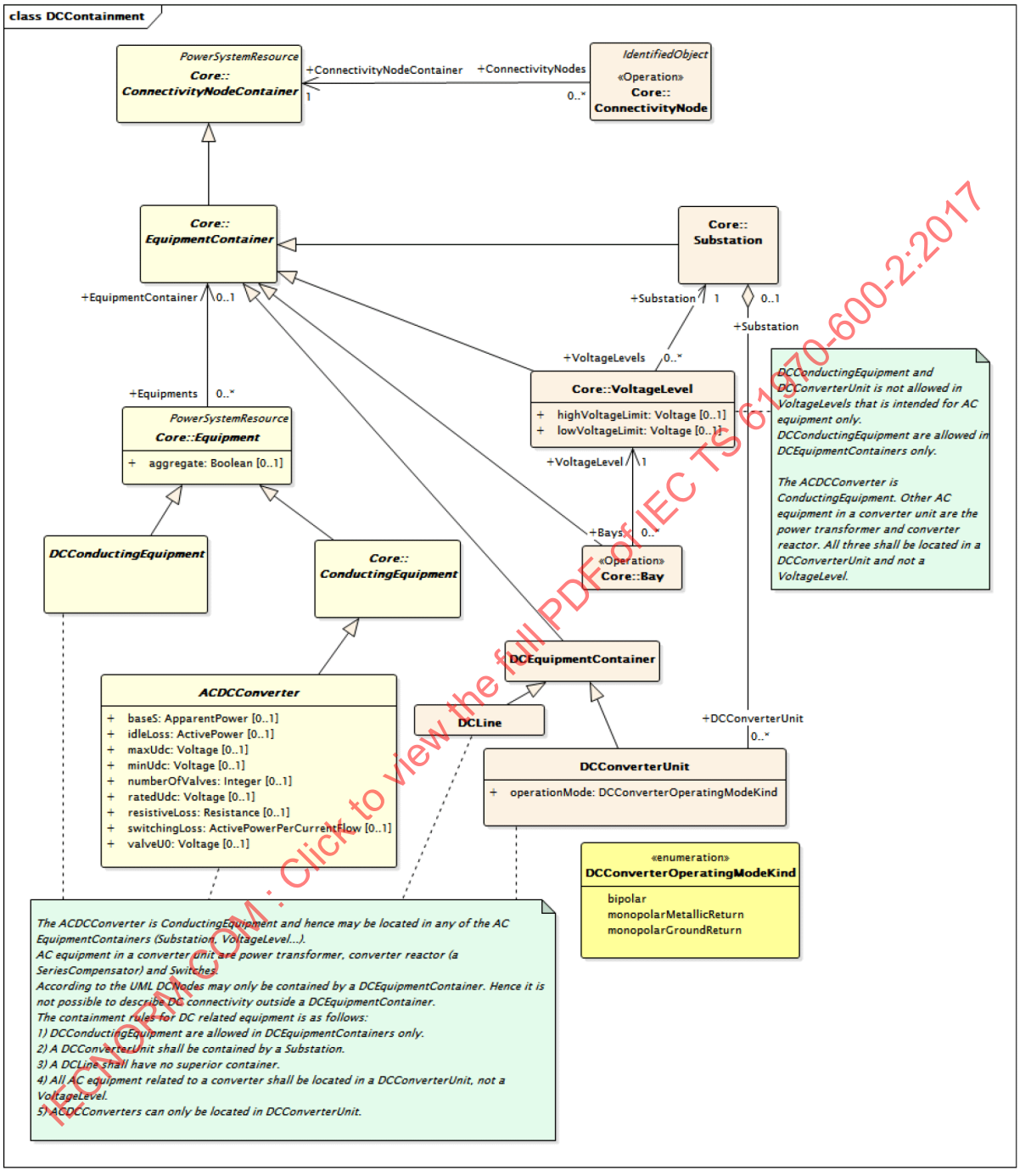


Figure 5 – diagram DCContainment

Figure 6 shows the diagram.

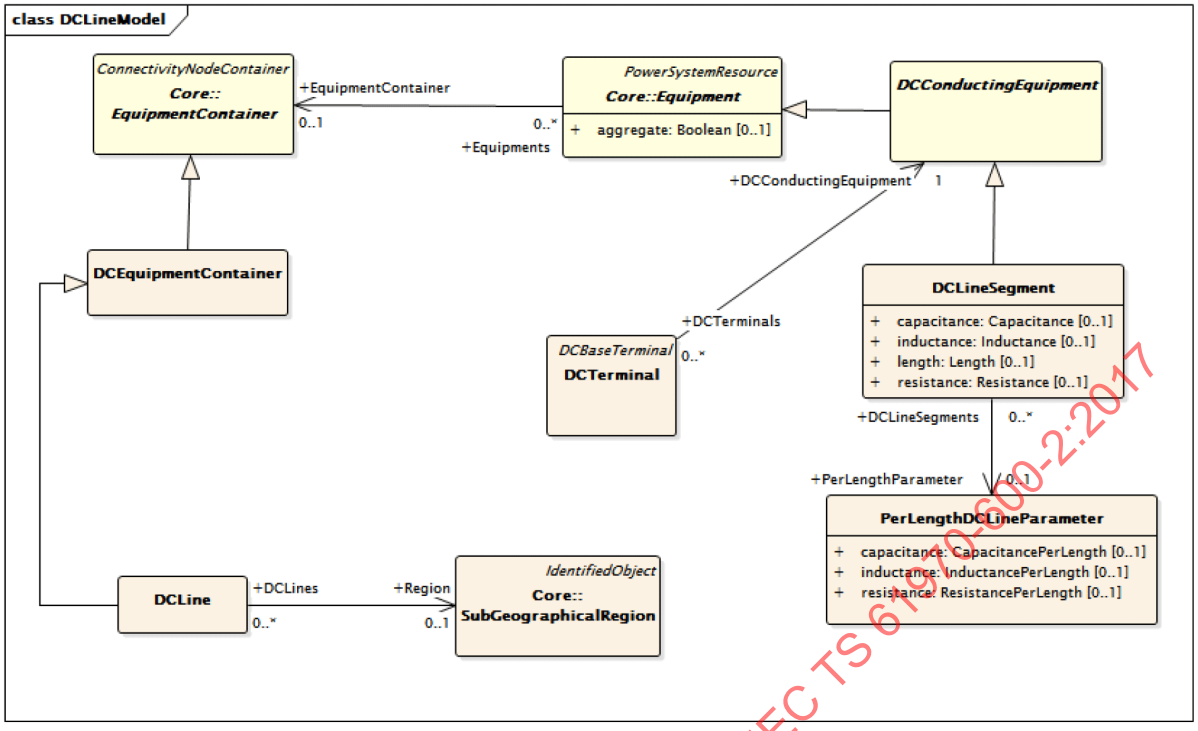


Figure 6 – diagram DCLineModel

Figure 7 shows the diagram.

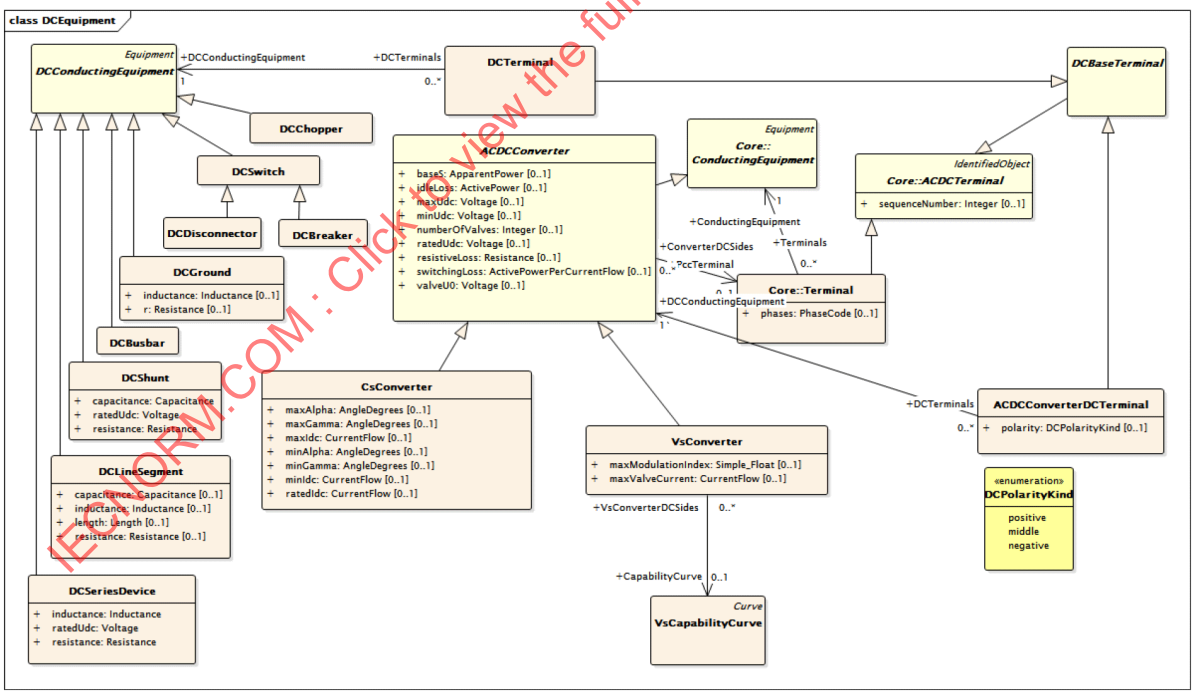


Figure 7 – diagram DCEquipment

This diagram shows classes used to model DC equipment.

Figure 8 shows the diagram.

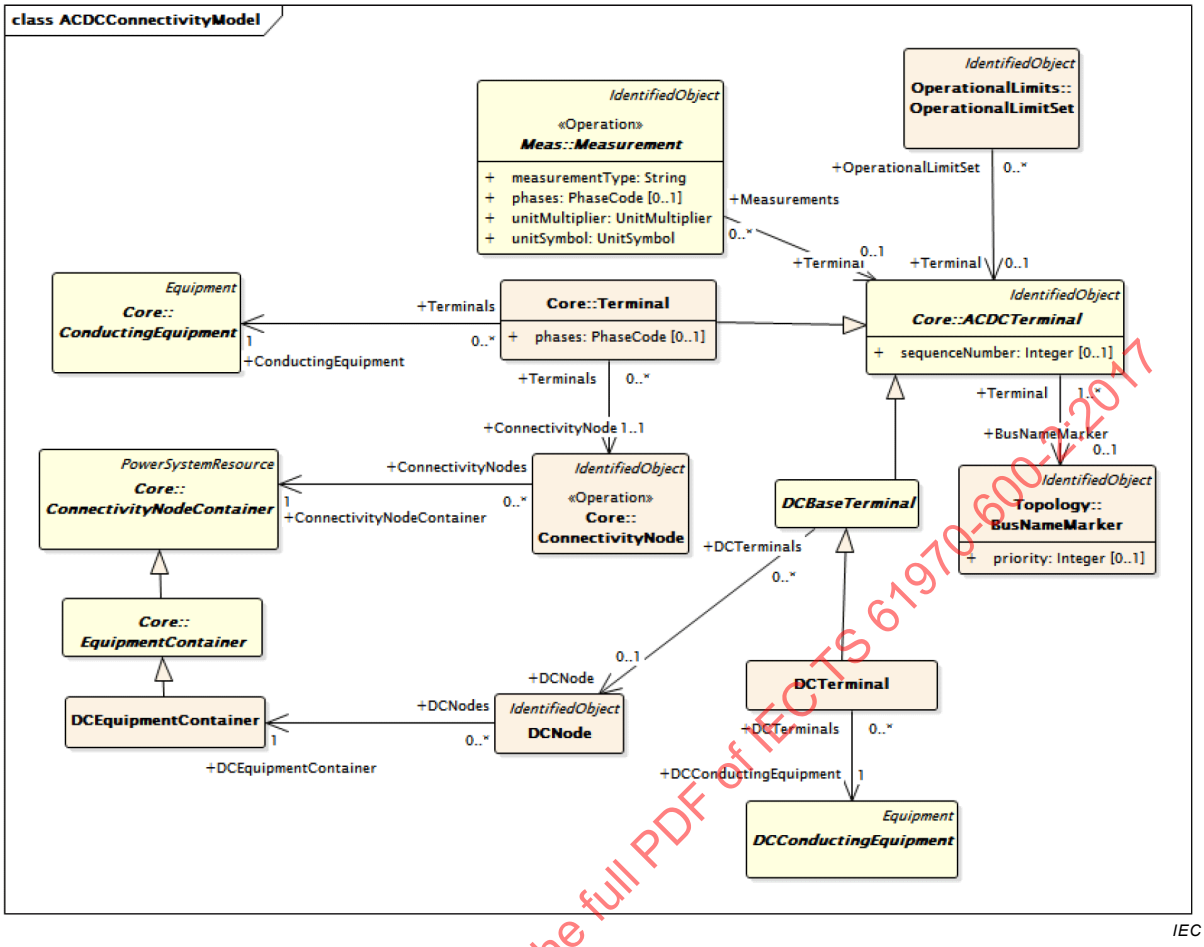


Figure 8 – diagram ACDCConnectivityModel

This diagram shows connectivity and topology model for both AC and DC networks.

6.3.2 ACDCConverter(Abstract)

A unit with valves for three phases, together with unit control equipment, essential protective and switching devices, DC storage capacitors, phase reactors and auxiliaries, if any, used for conversion.

OCL constraint: An ACDCConverter must be in a DCConveterUnit

The ACDCConverter is ConductingEquipment and hence may be located in any of the AC EquipmentContainers (Substation, VoltageLevel, etc.). AC equipment in a converter unit are power transformer, converter reactor (a SeriesCompensator) and Switches. According to the UML DCNodes may only be contained by a DCEquipmentContainer. Hence it is not possible to describe DC connectivity outside a DCEquipmentContainer. The containment rules for DC related equipment is as follows: 1) DCConductingEquipment are allowed in DCEquipmentContainers only. 2) A DCConverterUnit shall be contained by a Substation. 3) A DCLine shall have no superior container. 4) All AC equipment related to a converter shall be located in a DCConverterUnit, not a VoltageLevel. 5) ACDCConverters can only be located in DCConverterUnit.

Inheritance path: ->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 32 shows all attributes of ACDCConverter.

Table 32 – Attributes of DC::ACDCConverter

Attribute name	mult.	Attribute type	Description
baseS	0..1	ApparentPower	Base apparent power of the converter pole.
idleLoss	0..1	ActivePower	Active power loss in pole at no power transfer. Converter configuration data used in power flow.
maxUdc	0..1	Voltage	The maximum voltage on the DC side at which the converter should operate. Converter configuration data used in power flow.
minUdc	0..1	Voltage	Min allowed converter DC voltage. Converter configuration data used in power flow.
numberOfValves	0..1	Integer	Number of valves in the converter. Used in loss calculations.
ratedUdc	0..1	Voltage	Rated converter DC voltage, also called UdN. Converter configuration data used in power flow.
resistiveLoss	0..1	Resistance	Converter configuration data used in power flow. Refer to poleLossP.
switchingLoss	0..1	ActivePowerPerCurrentFlow	Switching losses, relative to the base apparent power 'baseS'. Refer to poleLossP.
valveU0	0..1	Voltage	Valve threshold voltage. Forward voltage drop when the valve is conducting. Used in loss calculations, i.e. the switchLoss depend on numberOfValves * valveU0.
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 33 shows all association ends of ACDCConverter with other classes.

Table 33 – Association ends of DC::ACDCConverter with other classes

mult.	mult.	Class type name	Description
PccTerminal	[0..1]	Terminal	All converters' DC sides linked to this point of common coupling terminal. Association Based On: DC:: ACDCConverter.ConverterDCSides[0..*] ----- Core:: Terminal.PccTerminal[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.3.3 ACDCConverterDCTerminal

A DC electrical connection point at the AC/DC converter. The AC/DC converter is electrically connected also to the AC side. The AC connection is inherited from the AC conducting equipment in the same way as any other AC equipment. The AC/DC converter DC terminal is separate from generic DC terminal to restrict the connection with the AC side to AC/DC converter and so that no other DC conducting equipment can be connected to the AC side.

Inheritance path: ->DCBaseTerminal->ACDCTerminal->IdentifiedObject

Table 34 shows all attributes of ACDCConverterDCTerminal.

Table 34 – Attributes of DC::ACDCConverterDCTerminal

Attribute name	mult.	Attribute type	Description
polarity	0..1	DCPolarityKind	Represents the normal network polarity condition.
sequenceNumber	0..1	Integer	see ACDCTerminal
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 35 shows all association ends of ACDCConverterDCTerminal with other classes.

Table 35 – Association ends of DC::ACDCConverterDCTerminal with other classes

mult.	mult.	Class type name	Description
DCConductingEquipment	[1..1]	ACDCConverter	Association Based On: DC:: ACDCConverterDCTerminal.DCTerminals[0..*] ----- DC:: ACDCConverter.DCConductingEquipment[1]
DCNode	0..1	DCNode	Inherited Association Based On: DC:: DCNode.DCNode[0..1] ----- DC:: DCBaseTerminal.DCTerminals[0..*]
BusNameMarker	0..1	BusNameMarker	Inherited Association Based On: Topology:: BusNameMarker.BusNameMarker[0..1] ----- Core:: ACDCTerminal.Terminal[1..*]

6.3.4 CsConverter

DC side of the current source converter (CSC).

Inheritance path: ->ACDCConverter->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 36 shows all attributes of CsConverter.

Table 36 – Attributes of DC::CsConverter

Attribute name	mult.	Attribute type	Description
maxAlpha	0..1	AngleDegrees	Maximum firing angle. CSC configuration data used in power flow.
maxGamma	0..1	AngleDegrees	Maximum extinction angle. CSC configuration data used in power flow.
maxIdc	0..1	CurrentFlow	The maximum direct current (Id) on the DC side at which the converter should operate. Converter configuration data use in power flow.
minAlpha	0..1	AngleDegrees	Minimum firing angle. CSC configuration data used in power flow.
minGamma	0..1	AngleDegrees	Minimum extinction angle. CSC configuration data used in power flow.
minIdc	0..1	CurrentFlow	The minimum direct current (Id) on the DC side at which the converter should operate. CSC configuration data used in power flow.
ratedIdc	0..1	CurrentFlow	Rated converter DC current, also called IdN. Converter configuration data used in power flow.
baseS	0..1	ApparentPower	see ACDCConverter
idleLoss	0..1	ActivePower	see ACDCConverter
maxUdc	0..1	Voltage	see ACDCConverter
minUdc	0..1	Voltage	see ACDCConverter
numberOfValves	0..1	Integer	see ACDCConverter
ratedUdc	0..1	Voltage	see ACDCConverter
resistiveLoss	0..1	Resistance	see ACDCConverter
switchingLoss	0..1	ActivePowerPerCurrentFlow	see ACDCConverter
valveU0	0..1	Voltage	see ACDCConverter
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 37 shows all association ends of ACDCConverter with other classes.

Table 37 – Association ends of DC::ACDCConverter with other classes

mult.	mult.	Class type name	Description
PccTerminal	0..1	Terminal	Inherited Association Based On: DC:: ACDCConverter.ConverterDCSides[0..*] ----- Core:: Terminal.PccTerminal[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.3.5 DCBaseTerminal(Abstract)

An electrical connection point at a piece of DC conducting equipment. DC terminals are connected at one physical DC node that may have multiple DC terminals connected. A DC node is similar to an AC connectivity node. The model enforces that DC connections are distinct from AC connections.

Inheritance path: ->ACDCTerminal->IdentifiedObject

Table 38 shows all attributes of DCBaseTerminal.

Table 38 – Attributes of DC::DCBaseTerminal

Attribute name	mult.	Attribute type	Description
sequenceNumber	0..1	Integer	see ACDCTerminal
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 39 shows all association ends of DCBaseTerminal with other classes.

Table 39 – Association ends of DC::DCBaseTerminal with other classes

mult.	mult.	Class type name	Description
DCNode	[0..1]	DCNode	Association Based On: DC:: DCNode.DCNode[0..1] ----- DC:: DCBaseTerminal.DCTerminals[0..*]
BusNameMarker	0..1	BusNameMarker	Inherited Association Based On: Topology:: BusNameMarker.BusNameMarker[0..1] ----- Core:: ACDCTerminal.Terminal[1..*]

6.3.6 DCBreaker

A breaker within a DC system.

Inheritance path: ->DCSwitch->DCConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 40 shows all attributes of DCBreaker.

Table 40 – Attributes of DC::DCBreaker

Attribute name	mult.	Attribute type	Description
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 41 shows all association ends of Equipment with other classes.

Table 41 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.3.7 DCBusbar

A busbar within a DC system.

Inheritance path: ->DCConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 42 shows all attributes of DCBusbar.

Table 42 – Attributes of DC::DCBusbar

Attribute name	mult.	Attribute type	Description
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 43 shows all association ends of Equipment with other classes.

Table 43 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	0..1	`EquipmentContainer`	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.3.8 DCChopper

Low resistance equipment used in the internal DC circuit to balance voltages. It has typically positive and negative pole terminals and a ground.

Inheritance path: ->DCConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 44 shows all attributes of DCChopper.

Table 44 – Attributes of DC::DCChopper

Attribute name	mult.	Attribute type	Description
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 45 shows all association ends of Equipment with other classes.

Table 45 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.3.9 DCConductingEquipment(Abstract)

The parts of the DC power system that are designed to carry current or that are conductively connected through DC terminals.

The ACDCConverter is ConductingEquipment and hence may be located in any of the AC EquipmentContainers (Substation, VoltageLevel etc.). AC equipment in a converter unit consists of power transformer, converter reactor (a SeriesCompensator) and Switches. According to the UML DCNodes may only be contained by a DCEquipmentContainer. Hence it is not possible to describe DC connectivity outside a DCEquipmentContainer. The containment rules for DC related equipment are as follows: 1) DCConductingEquipment are allowed in DCEquipmentContainers only. 2) A DCConverterUnit shall be contained by a Substation. 3) A DCLine shall have no superior container. 4) All AC equipment related to a converter shall be located in a DCConverterUnit, not a VoltageLevel. 5) ACDCConverters can only be located in DCConverterUnit.

Inheritance path: ->Equipment->PowerSystemResource->IdentifiedObject

Table 46 shows all attributes of DCConductingEquipment.

Table 46 – Attributes of DC::DCConductingEquipment

Attribute name	mult.	Attribute type	Description
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 47 shows all association ends of Equipment with other classes.

Table 47 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.3.10 DCConverterUnit

Indivisible operative unit comprising all equipment between the point of common coupling on the AC side and the point of common coupling – DC side, essentially one or more converters, together with one or more converter transformers, converter control equipment, essential protective and switching devices and auxiliaries, if any, used for conversion.

The ACDCConverter is ConductingEquipment and hence may be located in any of the AC EquipmentContainers (Substation, VoltageLevel etc.). AC equipment in a converter unit consists of power transformer, converter reactor (a SeriesCompensator) and Switches. According to the UML DCNodes may only be contained by a DCEquipmentContainer. Hence it is not possible to describe DC connectivity outside a DCEquipmentContainer. The containment rules for DC related equipment are as follows: 1) DCConductingEquipment are allowed in DCEquipmentContainers only. 2) A DCConverterUnit shall be contained by a Substation. 3) A DCLine shall have no superior container. 4) All AC equipment related to a converter shall be located in a DCConverterUnit, not a VoltageLevel. 5) ACDCConverters can only be located in DCConverterUnit.

Inheritance path: ->DCEquipmentContainer->EquipmentContainer->ConnectivityNodeContainer->PowerSystemResource->IdentifiedObject

Table 48 shows all attributes of DCConverterUnit.

Table 48 – Attributes of DC::DCConverterUnit

Attribute name	mult.	Attribute type	Description
operationMode	1..1	DCConverterOperatingModeKind	
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 49 shows all association ends of DCConverterUnit with other classes.

Table 49 – Association ends of DC::DCConverterUnit with other classes

mult.	mult.	Class type name	Description
Substation	[0..1]	Substation	Association Based On: DC:: DCConverterUnit.DCConverterUnit[0..*] ----- Core:: Substation.Substation[0..1]

6.3.11 DCDisconnector

A disconnector within a DC system.

Inheritance path: ->DCSwitch->DCConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 50 shows all attributes of DCDisconnector.

Table 50 – Attributes of DC::DCDisconnector

Attribute name	mult.	Attribute type	Description
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 51 shows all association ends of Equipment with other classes.

Table 51 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	0..1	`EquipmentContainer`	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] --- Core:: Equipment.Equipments[0..*]

6.3.12 DCEquipmentContainer

A modeling construct to provide a root class for containment of DC as well as AC equipment. The class differs from the EquipmentContaner for AC in that it may also contain DCNodes. Hence it can contain both AC and DC equipment.

Inheritance path: ->EquipmentContainer->ConnectivityNodeContainer->PowerSystemResource->IdentifiedObject

Table 52 shows all attributes of DCEquipmentContainer.

Table 52 – Attributes of DC::DCEquipmentContainer

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.3.13 DCGround

A ground within a DC system.

OCL constraint: A DCGround must be in a DCConverterUnit

Inheritance path: ->DCConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 53 shows all attributes of DCGround.

Table 53 – Attributes of DC::DCGround

Attribute name	mult.	Attribute type	Description
inductance	0..1	Inductance	Inductance to ground.
r	0..1	Resistance	Resistance to ground.
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 54 shows all association ends of Equipment with other classes.

Table 54 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] --- Core:: Equipment.Equipments[0..*]

6.3.14 DCLine

Overhead lines and/or cables connecting two or more HVDC substations.

The ACDCConverter is ConductingEquipment and hence may be located in any of the AC EquipmentContainers (Substation, VoltageLevel etc.). AC equipment in a converter unit consists of power transformer, converter reactor (a SeriesCompensator) and Switches. According to the UML DCNodes may only be contained by a DCEquipmentContainer. Hence it is not possible to describe DC connectivity outside a DCEquipmentContainer. The containment rules for DC related equipment are as follows: 1) DCConductingEquipment are allowed in DCEquipmentContainers only. 2) A DCConverterUnit shall be contained by a Substation. 3) A DCLine shall have no superior container. 4) All AC equipment related to a converter shall be located in a DCConverterUnit, not a VoltageLevel. 5) ACDCConverters can only be located in DCConverterUnit.

Inheritance path: ->DCEquipmentContainer->EquipmentContainer->ConnectivityNodeContainer->PowerSystemResource->IdentifiedObject

Table 55 shows all attributes of DCLine.

Table 55 – Attributes of DC::DCLine

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 56 shows all association ends of DCLine with other classes.

Table 56 – Association ends of DC::DCLine with other classes

mult.	mult.	Class type name	Description
Region	[0..1]	SubGeographicalRegion	Association Based On: DC:: DCLine.DCLines[0..*] ----- Core:: SubGeographicalRegion.Region[0..1]

6.3.15 DCLineSegment

A wire or combination of wires not insulated from one another, with consistent electrical characteristics, used to carry direct current between points in the DC region of the power system.

OCL constraint: A DCLineSegment must be contained in a DCLine

Inheritance path: ->DCConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 57 shows all attributes of DCLineSegment.

Table 57 – Attributes of DC::DCLineSegment

Attribute name	mult.	Attribute type	Description
capacitance	0..1	Capacitance	Capacitance of the DC line segment. Significant for cables only.
inductance	0..1	Inductance	Inductance of the DC line segment. Neglectable compared with DCSeriesDevice used for smoothing.
resistance	0..1	Resistance	Resistance of the DC line segment.
length	0..1	Length	Segment length for calculating line section capabilities.
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 58 shows all association ends of DCLineSegment with other classes.

Table 58 – Association ends of DC::DCLineSegment with other classes

mult.	mult.	Class type name	Description
PerLengthParameter	[0..1]	PerLengthDCLineParameter	Set of per-length parameters for this line segment. Association Based On: DC:: PerLengthDCLineParameter.PerLengthParameter[0..1] ----- DC:: DCLineSegment.DCLineSegments[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.3.16 DCNode

DC nodes are points where terminals of DC conducting equipment are connected together with zero impedance.

Inheritance path: ->IdentifiedObject

Table 59 shows all attributes of DCNode.

Table 59 – Attributes of DC::DCNode

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 60 shows all association ends of DCNode with other classes.

Table 60 – Association ends of DC::DCNode with other classes

mult.	mult.	Class type name	Description
DCEquipmentContainer	[1..1]	DCEquipmentContainer	Association Based On: DC:: DCNode.DCNodes[0..*] ----- DC:: DCEquipmentContainer.DCEquipmentContainer[1]

6.3.17 DCSeriesDevice

A series device within the DC system, typically a reactor used for filtering or smoothing. Needed for transient and short circuit studies.

Inheritance path: ->DCConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 61 shows all attributes of DCSeriesDevice.

Table 61 – Attributes of DC::DCSeriesDevice

Attribute name	mult.	Attribute type	Description
inductance	1..1	Inductance	Inductance of the device.
resistance	1..1	Resistance	Resistance of the DC device.
ratedUdc	1..1	Voltage	Rated DC device voltage. Converter configuration data used in power flow.
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 62 shows all association ends of Equipment with other classes.

Table 62 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] --- Core:: Equipment.Equipments[0..*]

6.3.18 DCShunt

A shunt device within the DC system, typically used for filtering. Needed for transient and short circuit studies.

Inheritance path: ->DCConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 63 shows all attributes of DCShunt.

Table 63 – Attributes of DC::DCShunt

Attribute name	mult.	Attribute type	Description
capacitance	1..1	Capacitance	Capacitance of the DC shunt.
resistance	1..1	Resistance	Resistance of the DC device.
ratedUdc	1..1	Voltage	Rated DC device voltage. Converter configuration data used in power flow.
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 64 shows all association ends of Equipment with other classes.

Table 64 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] --- Core:: Equipment.Equipments[0..*]

6.3.19 DCSwitch

A switch within the DC system.

Inheritance path: ->DCConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 65 shows all attributes of DCSwitch.

Table 65 – Attributes of DC::DCSwitch

Attribute name	mult.	Attribute type	Description
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 66 shows all association ends of Equipment with other classes.

Table 66 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: Equipment.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.3.20 DCTerminal

An electrical connection point to generic DC conducting equipment.

Inheritance path: ->DCBaseTerminal->ACDCTerminal->IdentifiedObject

Table 67 shows all attributes of DCTerminal.

Table 67 – Attributes of DC::DCTerminal

Attribute name	mult.	Attribute type	Description
sequenceNumber	0..1	Integer	see ACDCTerminal
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 68 shows all association ends of DCTerminal with other classes.

Table 68 – Association ends of DC::DCTerminal with other classes

mult.	mult.	Class type name	Description
DCConductingEquipment	[1..1]	DCConductingEquipment	Association Based On: DC:: DCConductingEquipment.DCConductingEquipment[1] ----- DC:: DCTerminal.DCTerminals[0..*]
DCNode	0..1	DCNode	Inherited Association Based On: DC:: DCNode.DCNode[0..1] ----- DC:: DCBaseTerminal.DCTerminals[0..*]
BusNameMarker	0..1	BusNameMarker	Inherited Association Based On: Topology:: BusNameMarker.BusNameMarker[0..1] ----- Core:: ACDCTerminal.Terminal[1..*]

6.3.21 PerLengthDCLineParameter

Table 69 shows all attributes of PerLengthDCLineParameter.

Table 69 – Attributes of DC::PerLengthDCLineParameter

Attribute name	mult.	Attribute type	Description
capacitance	0..1	CapacitancePerLength	Capacitance per unit of length of the DC line segment; significant for cables only.
inductance	0..1	InductancePerLength	Inductance per unit of length of the DC line segment.
resistance	0..1	ResistancePerLength	Resistance per length of the DC line segment.

6.3.22 VsCapabilityCurve

The P-Q capability curve for a voltage source converter, with P on x-axis and Qmin and Qmax on y1-axis and y2-axis.

Inheritance path ->Curve->IdentifiedObject

Table 70 shows all attributes of VsCapabilityCurve.

Table 70 – Attributes of DC::VsCapabilityCurve

Attribute name	mult.	Attribute type	Description
curveStyle	1..1	CurveStyle	see Curve
xUnit	1..1	UnitSymbol	see Curve
y1Unit	1..1	UnitSymbol	see Curve
y2Unit	0..1	UnitSymbol	see Curve
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.3.23 VsConverter

DC side of the voltage source converter (VSC).

Inheritance path: ->ACDCConverter->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 71 shows all attributes of VsConverter.

Table 71 – Attributes of DC::VsConverter

Attribute name	mult.	Attribute type	Description
maxModulationIndex	0..1	Simple_Float	The max quotient between the AC converter voltage (Uc) and DC voltage (Ud). A factor typically less than 1. VSC configuration data used in power flow.
maxValveCurrent	0..1	CurrentFlow	The maximum current through a valve. This current limit is the basis for calculating the capability diagram. VSC configuration data.
baseS	0..1	ApparentPower	see ACDCConverter
idleLoss	0..1	ActivePower	see ACDCConverter
maxUdc	0..1	Voltage	see ACDCConverter
minUdc	0..1	Voltage	see ACDCConverter
numberOfValves	0..1	Integer	see ACDCConverter
ratedUdc	0..1	Voltage	see ACDCConverter
resistiveLoss	0..1	Resistance	see ACDCConverter
switchingLoss	0..1	ActivePowerPerCurrentFlow	see ACDCConverter
valveU0	0..1	Voltage	see ACDCConverter
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 72 shows all association ends of VsConverter with other classes.

Table 72 – Association ends of DC::VsConverter with other classes

mult.	mult.	Class type name	Description
CapabilityCurve	[0..1]	VsCapabilityCurve	All converters with this capability curve. Association Based On: DC:: VsConverter.VsConverterDCSides[0..*] ----- DC:: VsCapabilityCurve.CapabilityCurve[0..1]
PccTerminal	0..1	Terminal	Inherited Association Based On: DC:: ACDCConverter.ConverterDCSides[0..*] ----- Core:: Terminal.PccTerminal[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.3.24 Enumerations

6.3.24.1 DCConverterOperatingModeKind

The operating mode of an HVDC bipole.

Literal	Description	Code
bipolar	Bipolar operation.	
monopolarMetallicReturn	Monopolar operation with metallic return	
monopolarGroundReturn	Monopolar operation with ground return	

6.3.24.2 DCPolarityKind

Polarity for DC circuits.

Literal	Description	Code
positive	Positive pole.	
middle	Middle pole, potentially grounded.	
negative	Negative pole.	

6.4 Topology

6.4.1 BusNameMarker

Used to apply user standard names to topology buses. Typically used for "bus/branch" case generation. Associated with one or more terminals that are normally connected with the bus name. The associated terminals are normally connected by non-retained switches. For a ring bus station configuration, all busbar terminals in the ring are typically associated. For a breaker and a half scheme, both busbars would normally be associated. For a ring bus, all

busbars would normally be associated. For a "straight" busbar configuration, normally only the main terminal at the busbar would be associated.

Inheritance path: ->IdentifiedObject

Table 73 shows all attributes of BusNameMarker.

Table 73 – Attributes of Topology::BusNameMarker

Attribute name	mult.	Attribute type	Description
priority	0..1	Integer	Priority of bus name marker for use as topology bus name. Use 0 for don t care. Use 1 for highest priority. Use 2 as priority is less than 1 and so on.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 74 shows all association ends of BusNameMarker with other classes.

Table 74 – Association ends of Topology::BusNameMarker with other classes

mult.	mult.	Class type name	Description
ReportingGroup	[0..1]	ReportingGroup	The bus name markers that belong to this reporting group. Association Based On: Topology:: BusNameMarker.BusNameMarker[0..*] ----- Core:: ReportingGroup.ReportingGroup[0..1]

6.5 Meas

6.5.1 General

Figure 9 shows the diagram.

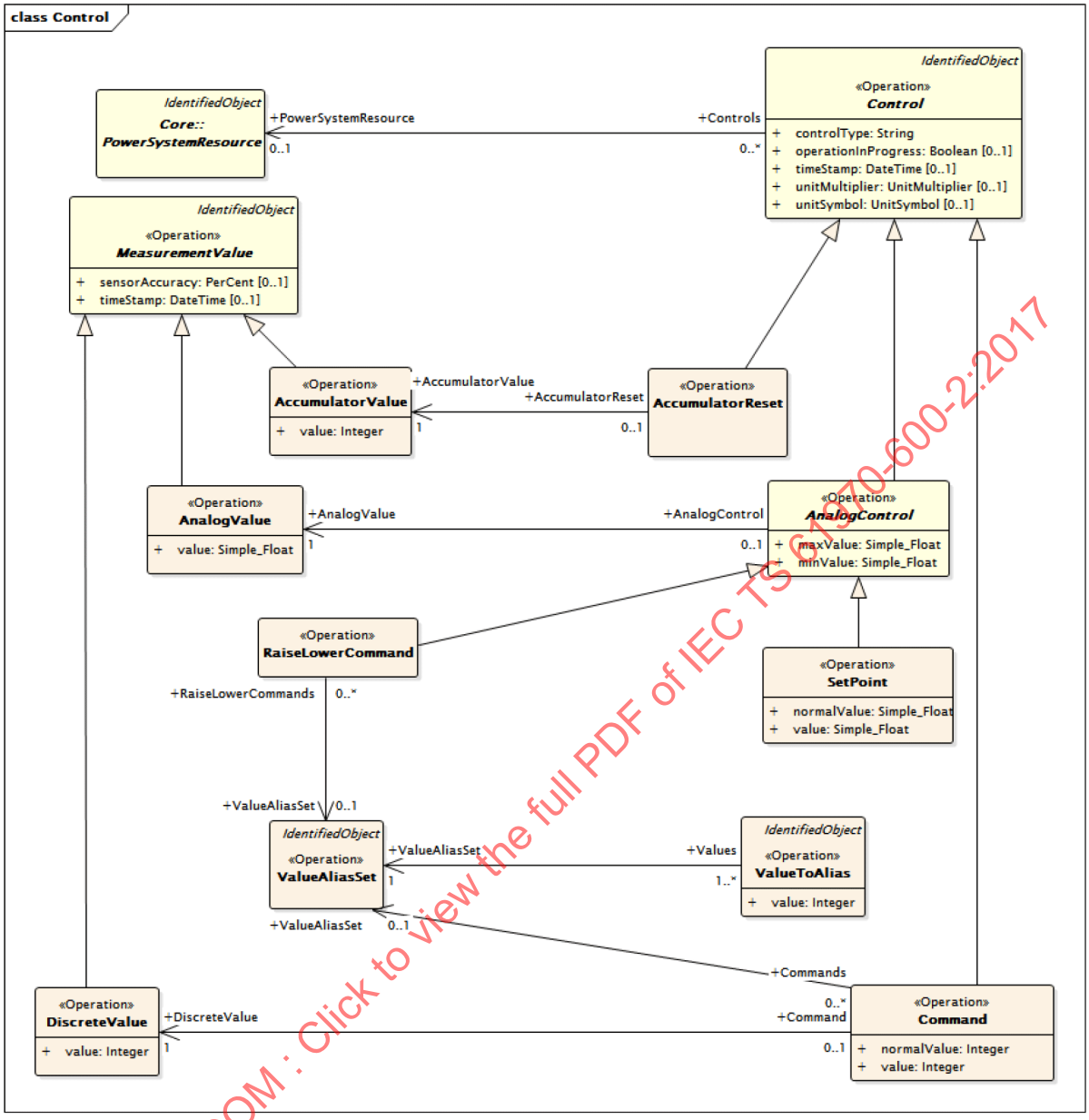
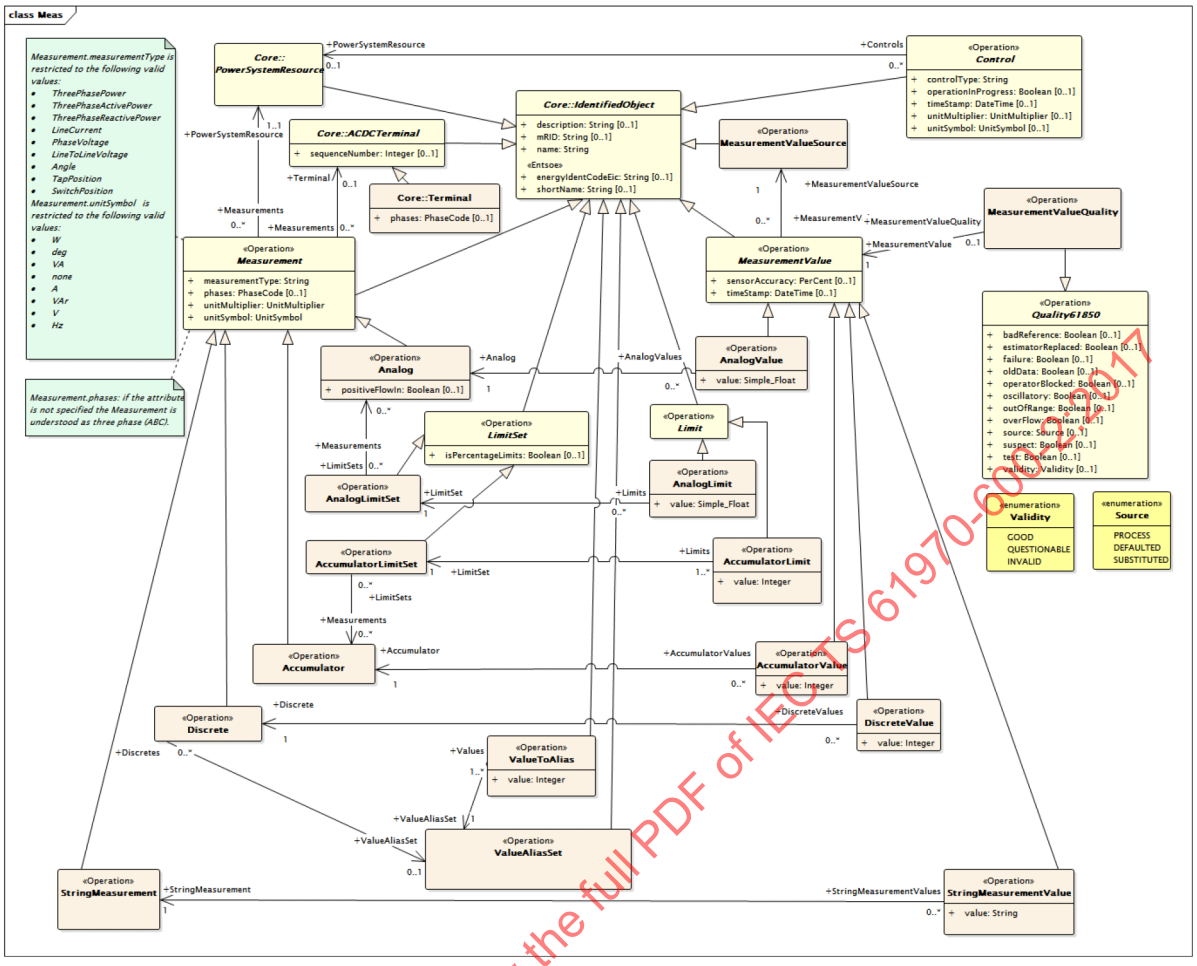


Figure 9 – diagram Control

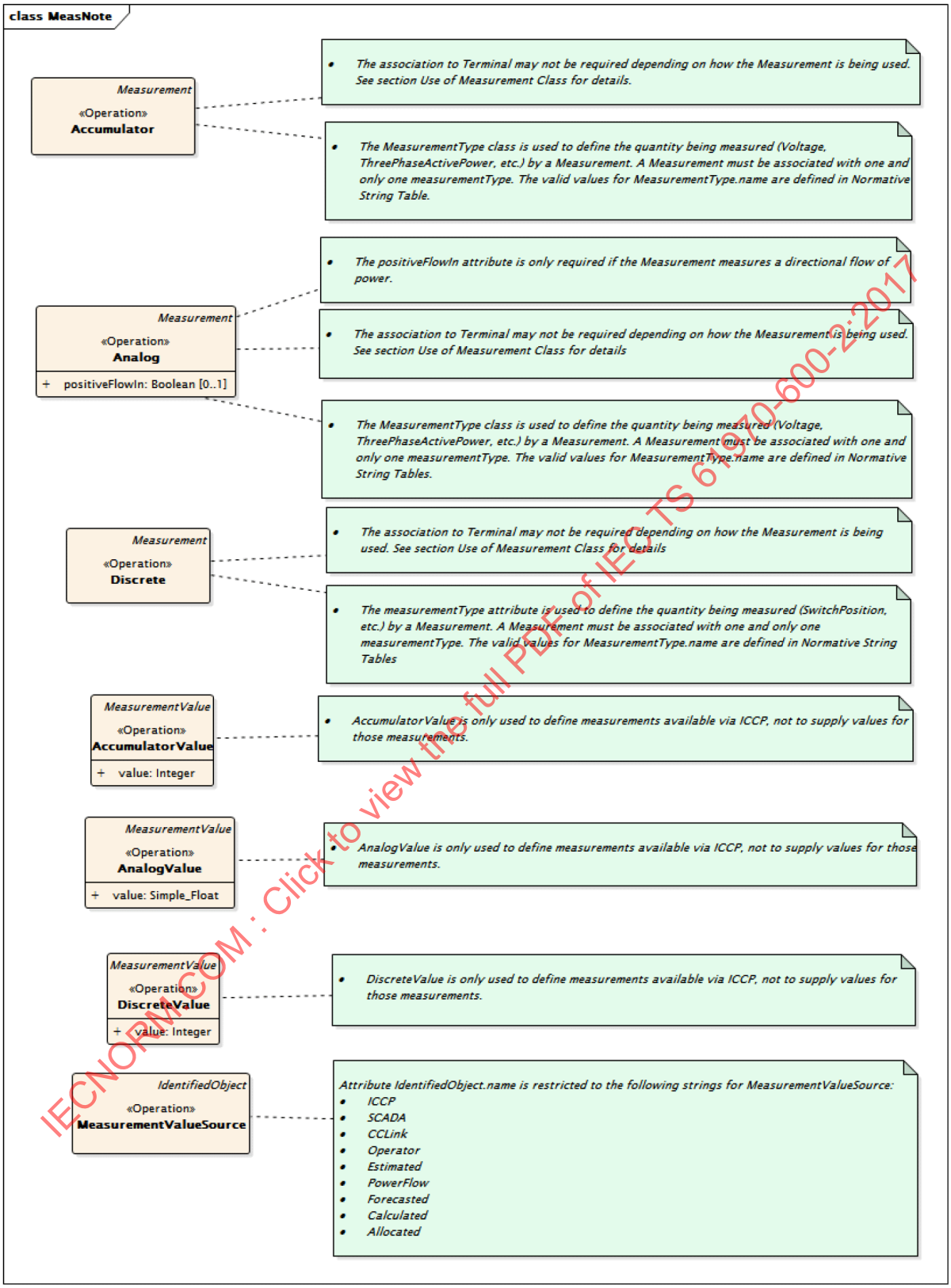
Figure 10 shows the diagram.



IEC

Figure 10 – diagram Meas

Figure 11 shows the diagram.



IEC

Figure 11 – diagram MeasNote

6.5.2 (Operation) Accumulator

Accumulator represents an accumulated (counted) Measurement, e.g. an energy value.

- The association to Terminal may not be required depending on how the Measurement is being used.
- The MeasurementType class is used to define the quantity being measured (Voltage, ThreePhaseActivePower, etc.) by a Measurement. A Measurement must be associated with one and only one measurementType. The valid values for MeasurementType.name are defined in Normative String Table.

Inheritance path: ->Measurement->IdentifiedObject

Table 75 shows all attributes of Accumulator.

Table 75 – Attributes of Meas::Accumulator

Attribute name	mult.	Attribute type	Description
measurementType	1..1	String	see Measurement
phases	0..1	PhaseCode	see Measurement
unitSymbol	1..1	UnitSymbol	see Measurement
unitMultiplier	1..1	UnitMultiplier	see Measurement
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 76 shows all association ends of Measurement with other classes.

Table 76 – Association ends of Meas::Measurement with other classes

mult.	mult.	Class type name	Description
Terminal (Operation)	0..1	ACDCTerminal	Inherited Association Based On: Meas:: Measurement.Measurements[0..*] ----- Core:: ACDCTerminal.Terminal[0..1]
PowerSystemResource (Operation)	1..1	PowerSystemResource	Inherited Association Based On: Core:: PowerSystemResource.PowerSystemResource[0..1] ----- Meas:: Measurement.Measurements[0..*]

6.5.3 (Operation) AccumulatorLimit

Limit values for Accumulator measurements.

Inheritance path: ->Limit->IdentifiedObject

Table 77 shows all attributes of AccumulatorLimit.

Table 77 – Attributes of Meas::AccumulatorLimit

Attribute name	mult.	Attribute type	Description
value (Operation)	1..1	Integer	The value to supervise against. The value is positive.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 78 shows all association ends of AccumulatorLimit with other classes.

Table 78 – Association ends of Meas::AccumulatorLimit with other classes

mult.	mult.	Class type name	Description
LimitSet (Operation)	[1..1]	AccumulatorLimitSet	The limit values used for supervision of Measurements. Association Based On: Meas:: AccumulatorLimitSet.LimitSet[1] ----- Meas:: AccumulatorLimit.Limits[1..*]

6.5.4 (Operation) AccumulatorLimitSet

An AccumulatorLimitSet specifies a set of Limits that are associated with an Accumulator measurement.

Inheritance path: ->LimitSet->IdentifiedObject

Table 79 shows all attributes of AccumulatorLimitSet.

Table 79 – Attributes of Meas::AccumulatorLimitSet

Attribute name	mult.	Attribute type	Description
isPercentageLimits	0..1	Boolean	see LimitSet
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 80 shows all association ends of AccumulatorLimitSet with other classes.

Table 80 – Association ends of Meas::AccumulatorLimitSet with other classes

mult.	mult.	Class type name	Description
Measurements (Operation)	[0..*]	Accumulator	A measurement may have zero or more limit ranges defined for it. Association Based On: Meas:: Accumulator.Measurements[0..*] ----- Meas:: AccumulatorLimitSet.LimitSets[0..*]

6.5.5 (Operation) AccumulatorReset

This command reset the counter value to zero.

Inheritance path: ->Control->IdentifiedObject

Table 81 shows all attributes of AccumulatorReset.

Table 81 – Attributes of Meas::AccumulatorReset

Attribute name	mult.	Attribute type	Description
controlType	1..1	String	see Control
operationInProgress	0..1	Boolean	see Control
timeStamp	0..1	DateTime	see Control
unitMultiplier	0..1	UnitMultiplier	see Control
unitSymbol	0..1	UnitSymbol	see Control
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 82 shows all association ends of AccumulatorReset with other classes.

Table 82 – Association ends of Meas::AccumulatorReset with other classes

mult.	mult.	Class type name	Description
AccumulatorValue (Operation)	[1..1]	AccumulatorValue	The accumulator value that is reset by the command. Association Based On: Meas:: AccumulatorReset.AccumulatorReset[0..1] ----- Meas:: AccumulatorValue.AccumulatorValue[1]
PowerSystemResource (Operation)	0..1	PowerSystemResource	Inherited Association Based On: Core:: PowerSystemResource.PowerSystemResource[0..1] ----- Meas:: Control.Controls[0..*]

6.5.6 (Operation) AccumulatorValue

AccumulatorValue represents an accumulated (counted) MeasurementValue.

- AccumulatorValue is only used to define measurements available via ICCP, not to supply values for those measurements.

Inheritance path: ->MeasurementValue->IdentifiedObject

Table 83 shows all attributes of AccumulatorValue.

Table 83 – Attributes of Meas::AccumulatorValue

Attribute name	mult.	Attribute type	Description
value (Operation)	1..1	Integer	The value to supervise. The value is positive.
timeStamp	0..1	DateTime	see MeasurementValue
sensorAccuracy	0..1	PerCent	see MeasurementValue
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 84 shows all association ends of AccumulatorValue with other classes.

Table 84 – Association ends of Meas::AccumulatorValue with other classes

mult.	mult.	Class type name	Description
Accumulator (Operation)	[1..1]	Accumulator	The values connected to this measurement. Association Based On: Meas:: Accumulator.Accumulator[1] ----- Meas:: AccumulatorValue.AccumulatorValues[0..*]
MeasurementValueSource (Operation)	1..1	MeasurementValueSource	Inherited Association Based On: Meas:: MeasurementValueSource.MeasurementValueSource[1] ----- Meas:: MeasurementValue.MeasurementValues[0..*]

6.5.7 (Operation) Analog

Analog represents an analog Measurement.

- The positiveFlowIn attribute is only required if the Measurement measures a directional flow of power.
- The association to Terminal may not be required depending on how the Measurement is being used.
- The MeasurementType class is used to define the quantity being measured (Voltage, ThreePhaseActivePower, etc.) by a Measurement. A Measurement must be associated with one and only one measurementType. The valid values for MeasurementType.name are defined in Normative String Tables.

Inheritance path: ->Measurement->IdentifiedObject

Table 85 shows all attributes of Analog.

Table 85 – Attributes of Meas::Analog

Attribute name	mult.	Attribute type	Description
positiveFlowIn (Operation)	0..1	Boolean	If true then this measurement is an active power, reactive power or current with the convention that a positive value measured at the Terminal means power is flowing into the related PowerSystemResource.
measurementType	1..1	String	see Measurement
phases	0..1	PhaseCode	see Measurement
unitSymbol	1..1	UnitSymbol	see Measurement
unitMultiplier	1..1	UnitMultiplier	see Measurement
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 86 shows all association ends of Measurement with other classes.

Table 86 – Association ends of Meas::Measurement with other classes

mult.	mult.	Class type name	Description
Terminal (Operation)	0..1	ACDCTerminal	Inherited Association Based On: Meas:: Measurement.Measurements[0..*] ----- Core:: ACDCTerminal.Terminal[0..1]
PowerSystemResource (Operation)	1..1	PowerSystemResource	Inherited Association Based On: Core:: PowerSystemResource.PowerSystemResource[0..1] ----- Meas:: Measurement.Measurements[0..*]

6.5.8 (Operation) AnalogControl(Abstract)

An analog control used for supervisory control.

Inheritance path: ->Control->IdentifiedObject

Table 87 shows all attributes of AnalogControl.

Table 87 – Attributes of Meas::AnalogControl

Attribute name	mult.	Attribute type	Description
maxValue (Operation)	1..1	Simple_Float	Normal value range maximum for any of the Control.value. Used for scaling, e.g. in bar graphs.
minValue (Operation)	1..1	Simple_Float	Normal value range minimum for any of the Control.value. Used for scaling, e.g. in bar graphs.
controlType	1..1	String	see Control
operationInProgress	0..1	Boolean	see Control
timeStamp	0..1	DateTime	see Control
unitMultiplier	0..1	UnitMultiplier	see Control
unitSymbol	0..1	UnitSymbol	see Control
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 88 shows all association ends of AnalogControl with other classes.

Table 88 – Association ends of Meas::AnalogControl with other classes

mult.	mult.	Class type name	Description
AnalogValue (Operation)	[1..1]	AnalogValue	The Control variable associated with the MeasurementValue. Association Based On: Meas:: AnalogValue.AnalogValue[1] ----- Meas:: AnalogControl.AnalogControl[0..1]
PowerSystemResource (Operation)	0..1	PowerSystemResource	Inherited Association Based On: Core:: PowerSystemResource.PowerSystemResource[0..1] ----- Meas:: Control.Controls[0..*]

6.5.9 (Operation) AnalogLimit

Limit values for Analog measurements.

Inheritance path: ->Limit->IdentifiedObject

Table 89 shows all attributes of AnalogLimit.

Table 89 – Attributes of Meas::AnalogLimit

Attribute name	mult.	Attribute type	Description
value (Operation)	1..1	Simple_Float	The value to supervise against.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 90 shows all association ends of AnalogLimit with other classes.

Table 90 – Association ends of Meas::AnalogLimit with other classes

mult.	mult.	Class type name	Description
LimitSet (Operation)	[1..1]	AnalogLimitSet	The limit values used for supervision of Measurements. Association Based On: Meas:: AnalogLimitSet.LimitSet[1] ----- Meas:: AnalogLimit.Limits[0..*]

6.5.10 (Operation) AnalogLimitSet

An AnalogLimitSet specifies a set of Limits that are associated with an Analog measurement.

Inheritance path: ->LimitSet->IdentifiedObject

Table 91 shows all attributes of AnalogLimitSet.

Table 91 – Attributes of Meas::AnalogLimitSet

Attribute name	mult.	Attribute type	Description
isPercentageLimits	0..1	Boolean	see LimitSet
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 92 shows all association ends of AnalogLimitSet with other classes.

Table 92 – Association ends of Meas::AnalogLimitSet with other classes

mult.	mult.	Class type name	Description
Measurements (Operation)	[0..*]	Analog	A measurement may have zero or more limit ranges defined for it. Association Based On: Meas:: Analog.Measurements[0..*] ----- Meas:: AnalogLimitSet.LimitSets[0..*]

6.5.11 (Operation) AnalogValue

AnalogValue represents an analog MeasurementValue.

- AnalogValue is only used to define measurements available via ICCP, not to supply values for those measurements.

Inheritance path: ->MeasurementValue->IdentifiedObject

Table 93 shows all attributes of AnalogValue.

Table 93 – Attributes of Meas::AnalogValue

Attribute name	mult.	Attribute type	Description
value (Operation)	1..1	Simple_Float	The value to supervise.
timeStamp	0..1	DateTime	see MeasurementValue
sensorAccuracy	0..1	PerCent	see MeasurementValue
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 94 shows all association ends of AnalogValue with other classes.

Table 94 – Association ends of Meas::AnalogValue with other classes

mult.	mult.	Class type name	Description
Analog (Operation)	[1..1]	Analog	The values connected to this measurement. Association Based On: Meas:: Analog.Analog[1] ----- Meas:: AnalogValue.AnalogValues[0..*]
MeasurementValueSource (Operation)	1..1	MeasurementValueSource	Inherited Association Based On: Meas:: MeasurementValueSource.MeasurementValueSource[1] ----- Meas:: MeasurementValue.MeasurementValues[0..*]

6.5.12 (Operation) Command

A Command is a discrete control used for supervisory control.

Inheritance path: ->Control->IdentifiedObject

Table 95 shows all attributes of Command.

Table 95 – Attributes of Meas::Command

Attribute name	mult.	Attribute type	Description
normalValue (Operation)	1..1	Integer	Normal value for Control.value e.g. used for percentage scaling.
value (Operation)	1..1	Integer	The value representing the actuator output.
controlType	1..1	String	see Control
operationInProgress	0..1	Boolean	see Control
timeStamp	0..1	DateTime	see Control
unitMultiplier	0..1	UnitMultiplier	see Control
unitSymbol	0..1	UnitSymbol	see Control
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 96 shows all association ends of Command with other classes.

Table 96 – Association ends of Meas::Command with other classes

mult.	mult.	Class type name	Description
DiscreteValue (Operation)	[1..1]	DiscreteValue	The Control variable associated with the MeasurementValue. Association Based On: Meas:: DiscreteValue.DiscreteValue[1] ----- Meas:: Command.Command[0..1]
ValueAliasSet (Operation)	[0..1]	ValueAliasSet	The ValueAliasSet used for translation of a Control value to a name. Association Based On: Meas:: Command.Commands[0..*] ----- Meas:: ValueAliasSet.ValueAliasSet[0..1]
PowerSystemResource (Operation)	0..1	PowerSystemResource	Inherited Association Based On: Core:: PowerSystemResource.PowerSystemResource[0..1] ----- Meas:: Control.Controls[0..*]

6.5.13 (Operation) Control(Abstract)

Control is used for supervisory/device control. It represents control outputs that are used to change the state in a process, e.g. close or open breaker, a set point value or a raise lower command.

Inheritance path: ->IdentifiedObject

Table 97 shows all attributes of Control.

Table 97 – Attributes of Meas::Control

Attribute name	mult.	Attribute type	Description
controlType (Operation)	1..1	String	Specifies the type of Control, e.g. BreakerOn/Off, GeneratorVoltageSetPoint, TieLineFlow etc. The ControlType.name shall be unique among all specified types and describe the type.
operationInProgress (Operation)	0..1	Boolean	Indicates that a client is currently sending control commands that has not completed.
timeStamp (Operation)	0..1	DateTime	The last time a control output was sent.
unitMultiplier (Operation)	0..1	UnitMultiplier	The unit multiplier of the controlled quantity.
unitSymbol (Operation)	0..1	UnitSymbol	The unit of measure of the controlled quantity.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 98 shows all association ends of Control with other classes.

Table 98 – Association ends of Meas::Control with other classes

mult.	mult.	Class type name	Description
PowerSystemResource (Operation)	[0..1]	PowerSystemResource	The controller outputs used to actually govern a regulating device, e.g. the magnetization of a synchronous machine or capacitor bank breaker actuator. Association Based On: Core:: PowerSystemResource.PowerSystemResource[0..1] ----- Meas:: Control.Controls[0..*]

6.5.14 (Operation) Discrete

Discrete represents a discrete Measurement, i.e. a Measurement representing discrete values, e.g. a Breaker position.

- The measurementType attribute is used to define the quantity being measured (SwitchPosition, etc.) by a Measurement. A Measurement shall be associated with one and only one measurementType. The valid values for MeasurementType.name are defined in Normative String Tables
- The association to Terminal may not be required depending on how the Measurement is being used. See section Use of Measurement Class for details

Inheritance path: ->Measurement->IdentifiedObject

Table 99 shows all attributes of Discrete.

Table 99 – Attributes of Meas::Discrete

Attribute name	mult.	Attribute type	Description
measurementType	1..1	String	see Measurement
phases	0..1	PhaseCode	see Measurement
unitSymbol	1..1	UnitSymbol	see Measurement
unitMultiplier	1..1	UnitMultiplier	see Measurement
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 100 shows all association ends of Discrete with other classes.

Table 100 – Association ends of Meas::Discrete with other classes

mult.	mult.	Class type name	Description
ValueAliasSet (Operation)	[0..1]	ValueAliasSet	The ValueAliasSet used for translation of a MeasurementValue.value to a name. Association Based On: Meas:: ValueAliasSet.ValueAliasSet[0..1] ----- Meas:: Discrete.Discretes[0..*]
Terminal (Operation)	0..1	ACDCTerminal	Inherited Association Based On: Meas:: Measurement.Measurements[0..*] ----- Core:: ACDCTerminal.Terminal[0..1]
PowerSystemResource (Operation)	1..1	PowerSystemResource	Inherited Association Based On: Core:: PowerSystemResource.PowerSystemResource[0..1] ----- Meas:: Measurement.Measurements[0..*]

6.5.15 (Operation) DiscreteValue

DiscreteValue represents a discrete MeasurementValue.

- DiscreteValue is only used to define measurements available via ICCP, not to supply values for those measurements.

Inheritance path: ->MeasurementValue->IdentifiedObject

Table 101 shows all attributes of DiscreteValue.

Table 101 – Attributes of Meas::DiscreteValue

Attribute name	mult.	Attribute type	Description
value (Operation)	1..1	Integer	The value to supervise.
timeStamp	0..1	DateTime	see MeasurementValue
sensorAccuracy	0..1	PerCent	see MeasurementValue
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 102 shows all association ends of DiscreteValue with other classes.

Table 102 – Association ends of Meas::DiscreteValue with other classes

mult.	mult.	Class type name	Description
Discrete (Operation)	[1..1]	Discrete	The values connected to this measurement. Association Based On: Meas:: Discrete.Discrete[1] ----- Meas:: DiscreteValue.DiscreteValues[0..*]
MeasurementValueSource (Operation)	1..1	MeasurementValueSource	Inherited Association Based On: Meas:: MeasurementValueSource.MeasurementValueSource[1] ----- Meas:: MeasurementValue.MeasurementValues[0..*]

6.5.16 (Operation) Limit(Abstract)

Specifies one limit value for a Measurement. A Measurement typically has several limits that are kept together by the LimitSet class. The actual meaning and use of a Limit instance (i.e., if it is an alarm or warning limit or if it is a high or low limit) is not captured in the Limit class. However the name of a Limit instance may indicate both meaning and use.

Inheritance path: ->IdentifiedObject

Table 103 shows all attributes of Limit.

Table 103 – Attributes of Meas::Limit

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.5.17 (Operation) LimitSet(Abstract)

Specifies a set of Limits that are associated with a Measurement. A Measurement may have several LimitSets corresponding to seasonal or other changing conditions. The condition is captured in the name and description attributes. The same LimitSet may be used for several Measurements. In particular percentage limits are used this way.

Inheritance path: ->IdentifiedObject

Table 104 ashows all attributes of LimitSet.

Table 104 – Attributes of Meas::LimitSet

Attribute name	mult.	Attribute type	Description
isPercentageLimits (Operation)	0..1	Boolean	Tells if the limit values are in percentage of normalValue or the specified Unit for Measurements and Controls.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.5.18 (Operation) Measurement(Abstract)

A Measurement represents any measured, calculated or non-measured non-calculated quantity. Any piece of equipment may contain Measurements, e.g. a substation may have temperature measurements and door open indications, a transformer may have oil temperature and tank pressure measurements, a bay may contain a number of power flow measurements and a Breaker may contain a switch status measurement. The PSR – Measurement association is intended to capture this use of Measurement and is included in the naming hierarchy based on EquipmentContainer. The naming hierarchy typically has Measurements as leafs, e.g. Substation-VoltageLevel-Bay-Switch-Measurement. Some Measurements represent quantities related to a particular sensor location in the network, e.g. a voltage transformer (PT) at a busbar or a current transformer (CT) at the bar between a breaker and an isolator. The sensing position is not captured in the PSR – Measurement association. Instead it is captured by the Measurement – Terminal association that is used to define the sensing location in the network topology. The location is defined by the connection of the Terminal to ConductingEquipment. If both a Terminal and PSR are associated, and the PSR is of type ConductingEquipment, the associated Terminal should belong to that ConductingEquipment instance. When the sensor location is needed both Measurement-PSR and Measurement-Terminal are used. The Measurement-Terminal association is never used alone.

OCL constraint: If a Terminal exists its ConductingEquipment must be the same as PSR

OCL constraint: measurementType is restricted to the values: ThreePhasePower, ThreePhaseActivePower, ThreePhaseReactivePower, LineCurrent, PhaseVoltage, LineToLineVoltage, Angle, TapPosition, SwitchPosition

OCL constraint: unitSymbol is limited to W, deg, VA, none, A, VAr, V, Hz

Measurement.measurementType is restricted to the following valid values:

- ThreePhasePower
- ThreePhaseActivePower
- ThreePhaseReactivePower

- LineCurrent
- PhaseVoltage
- LineToLineVoltage
- Angle
- TapPosition
- SwitchPosition

Measurement.unitSymbol is restricted to the following valid values:

- W
- deg
- VA
- none
- A
- VAr
- V
- Hz

Measurement.phases: if the attribute is not specified the Measurement is understood as three phase (ABC).

Inheritance path: ->IdentifiedObject

Table 105 shows all attributes of Measurement.

Table 105 – Attributes of Meas::Measurement

Attribute name	mult.	Attribute type	Description
measurementType (Operation)	1..1	String	Specifies the type of measurement. For example, this specifies if the measurement represents an indoor temperature, outdoor temperature, bus voltage, line flow, etc.
phases (Operation)	0..1	PhaseCode	Indicates to which phases the measurement applies and avoids the need to use 'measurementType' to also encode phase information (which would explode the types). The phase information in Measurement, along with 'measurementType' and 'phases' uniquely defines a Measurement for a device, based on normal network phase. Their meaning will not change when the computed energizing phasing is changed due to jumpers or other reasons. If the attribute is missing three phases (ABC) shall be assumed.
unitSymbol (Operation)	1..1	UnitSymbol	The unit of measure of the measured quantity.
unitMultiplier (Operation)	1..1	UnitMultiplier	The unit multiplier of the measured quantity.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 106 shows all association ends of Measurement with other classes.

Table 106 – Association ends of Meas::Measurement with other classes

mult.	mult.	Class type name	Description
Terminal (Operation)	[0..1]	ACDCTerminal	One or more measurements may be associated with a terminal in the network. Association Based On: Meas:: Measurement.Measurements[0..*] ----- Core:: ACDCTerminal.Terminal[0..1]
PowerSystemResource (Operation)	[1..1]	PowerSystemResource	The measurements associated with this power system resource. Association Based On: Core:: PowerSystemResource.PowerSystemResource[0..1] ----- Meas:: Measurement.Measurements[0..*]

6.5.19 (Operation) MeasurementValue(Abstract)

The current state for a measurement. A state value is an instance of a measurement from a specific source. Measurements can be associated with many state values, each representing a different source for the measurement.

Inheritance path: ->IdentifiedObject

Table 107 shows all attributes of MeasurementValue.

Table 107 – Attributes of Meas::MeasurementValue

Attribute name	mult.	Attribute type	Description
timeStamp (Operation)	0..1	DateTime	The time when the value was last updated
sensorAccuracy (Operation)	0..1	PerCent	The limit, expressed as a percentage of the sensor maximum, that errors will not exceed when the sensor is used under reference conditions.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 108 shows all association ends of MeasurementValue with other classes.

Table 108 – Association ends of Meas::MeasurementValue with other classes

mult.	mult.	Class type name	Description
MeasurementValueSource (Operation)	[1..1]	MeasurementValueSource	The MeasurementValues updated by the source. Association Based On: Meas:: MeasurementValueSource.MeasurementValueSource[1] ----- Meas:: MeasurementValue.MeasurementValues[0..*]

6.5.20 (Operation) MeasurementValueQuality

Measurement quality flags. Bits 0-10 are defined for substation automation in IEC 61850-7-3/AMD1[2]. Bits 11-15 are reserved for future expansion by that document. Bits 16-31 are reserved for EMS applications.

Inheritance path: ->Quality61850

Table 109 shows all attributes of MeasurementValueQuality.

Table 109 – Attributes of Meas::MeasurementValueQuality

Attribute name	mult.	Attribute type	Description
badReference	0..1	Boolean	see Quality61850
estimatorReplaced	0..1	Boolean	see Quality61850
failure	0..1	Boolean	see Quality61850
oldData	0..1	Boolean	see Quality61850
operatorBlocked	0..1	Boolean	see Quality61850
oscillatory	0..1	Boolean	see Quality61850
outOfRange	0..1	Boolean	see Quality61850
overFlow	0..1	Boolean	see Quality61850
source	0..1	Source	see Quality61850
suspect	0..1	Boolean	see Quality61850
test	0..1	Boolean	see Quality61850
validity	0..1	Validity	see Quality61850

Table 110 shows all association ends of MeasurementValueQuality with other classes.

Table 110 – Association ends of Meas::MeasurementValueQuality with other classes

mult.	mult.	Class type name	Description
MeasurementValue (Operation)	[1..1]	MeasurementValue	A MeasurementValue has a MeasurementValueQuality associated with it. Association Based On: Meas:: MeasurementValueQuality.MeasurementValueQuality[0..1] - --- Meas:: MeasurementValue.MeasurementValue[1]

6.5.21 (Operation) MeasurementValueSource

MeasurementValueSource describes the alternative sources updating a MeasurementValue. User conventions for how to use the MeasurementValueSource attributes are described in the introduction to IEC 61970-301.

Attribute IdentifiedObject.name is restricted to the following strings for MeasurementValueSource:

---

2 To be published. Stage at the time of publication: IEC/AFDIS 61850-7-3/AMD1:2017.

- ICCP
- SCADA
- CCLink
- Operator
- Estimated
- PowerFlow
- Forecasted
- Calculated
- Allocated

Inheritance path: ->IdentifiedObject

Table 111 shows all attributes of MeasurementValueSource.

Table 111 – Attributes of Meas::MeasurementValueSource

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.5.22 (Operation) Quality61850(Abstract)

Quality flags in this class are as defined in IEC 61850, except for estimatorReplaced, which has been included in this class for convenience.

Table 112 shows all attributes of Quality61850.

Table 112 – Attributes of Meas::Quality61850

Attribute name	mult.	Attribute type	Description
badReference (Operation)	0..1	Boolean	Measurement value may be incorrect due to a reference being out of calibration.
estimatorReplaced (Operation)	0..1	Boolean	Value has been replaced by State Estimator. estimatorReplaced is not an IEC61850 quality bit but has been put in this class for convenience.
failure (Operation)	0..1	Boolean	This identifier indicates that a supervision function has detected an internal or external failure, e.g. communication failure.
oldData (Operation)	0..1	Boolean	Measurement value is old and possibly invalid, as it has not been successfully updated during a specified time interval.
operatorBlocked (Operation)	0..1	Boolean	Measurement value is blocked and hence unavailable for transmission.
oscillatory (Operation)	0..1	Boolean	To prevent some overload of the communication it is sensible to detect and suppress oscillating (fast changing) binary inputs. If a signal changes in a defined time (tosc) twice in the same direction (from 0 to 1 or from 1 to 0) then oscillation is detected and the detail quality identifier "oscillatory" is set. If it is detected a configured numbers of transient changes could be passed by. In this time the validity status "questionable" is set. If after this defined numbers of changes the signal is still in the oscillating state the value shall be set either to the opposite state of the previous stable value or to a defined default value. In this case the validity status "questionable" is reset and "invalid" is set as long as the signal is oscillating. If it is configured such that no transient changes should be passed by then the validity status "invalid" is set immediately in addition to the detail quality identifier "oscillatory" (used for status information only).
outOfRange (Operation)	0..1	Boolean	Measurement value is beyond a predefined range of value.
overFlow (Operation)	0..1	Boolean	Measurement value is beyond the capability of being represented properly. For example, a counter value overflows from maximum count back to a value of zero.
source (Operation)	0..1	Source	Source gives information related to the origin of a value. The value may be acquired from the process, defaulted or substituted.
suspect (Operation)	0..1	Boolean	A correlation function has detected that the value is not consitent with other values. Typically set by a network State Estimator.
test (Operation)	0..1	Boolean	Measurement value is transmitted for test purposes.
validity (Operation)	0..1	Validity	Validity of the measurement value.

6.5.23 (Operation) RaiseLowerCommand

An analog control that increase or decrease a set point value with pulses.

Inheritance path: ->AnalogControl->Control->IdentifiedObject

Table 113 shows all attributes of RaiseLowerCommand.

Table 113 – Attributes of Meas::RaiseLowerCommand

Attribute name	mult.	Attribute type	Description
maxValue	1..1	Simple_Float	see AnalogControl
minValue	1..1	Simple_Float	see AnalogControl
controlType	1..1	String	see Control
operationInProgress	0..1	Boolean	see Control
timeStamp	0..1	DateTime	see Control
unitMultiplier	0..1	UnitMultiplier	see Control
unitSymbol	0..1	UnitSymbol	see Control
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 114 shows all association ends of RaiseLowerCommand with other classes.

Table 114 – Association ends of Meas::RaiseLowerCommand with other classes

mult.	mult.	Class type name	Description
ValueAliasSet (Operation)	[0..1]	ValueAliasSet	The ValueAliasSet used for translation of a Control value to a name. Association Based On: Meas:: RaiseLowerCommand.RaiseLowerCommands[0..*] - ---- Meas:: ValueAliasSet.ValueAliasSet[0..1]
AnalogValue (Operation)	1..1	AnalogValue	Inherited Association Based On: Meas:: AnalogValue.AnalogValue[1] ----- Meas:: AnalogControl.AnalogControl[0..1]
PowerSystemResource (Operation)	0..1	PowerSystemResource	Inherited Association Based On: Core:: PowerSystemResource.PowerSystemResource[0..1] ----- Meas:: Control.Controls[0..*]

6.5.24 (Operation) SetPoint

An analog control that issue a set point value.

Inheritance path: ->AnalogControl->Control->IdentifiedObject

Table 115 shows all attributes of SetPoint.

Table 115 – Attributes of Meas::SetPoint

Attribute name	mult.	Attribute type	Description
normalValue (Operation)	1..1	Simple_Float	Normal value for Control.value e.g. used for percentage scaling.
value (Operation)	1..1	Simple_Float	The value representing the actuator output.
maxValue	1..1	Simple_Float	see AnalogControl
minValue	1..1	Simple_Float	see AnalogControl
controlType	1..1	String	see Control
operationInProgress	0..1	Boolean	see Control
timeStamp	0..1	DateTime	see Control
unitMultiplier	0..1	UnitMultiplier	see Control
unitSymbol	0..1	UnitSymbol	see Control
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 116 shows all association ends of AnalogControl with other classes.

Table 116 – Association ends of Meas::AnalogControl with other classes

mult.	mult.	Class type name	Description
AnalogValue (Operation)	1..1	AnalogValue	Inherited Association Based On: Meas:: AnalogValue.AnalogValue[1] ----- Meas:: AnalogControl.AnalogControl[0..1]
PowerSystemResource (Operation)	0..1	PowerSystemResource	Inherited Association Based On: Core:: PowerSystemResource.PowerSystemResource[0..1] ----- Meas:: Control.Controls[0..*]

6.5.25 (Operation) StringMeasurement

StringMeasurement represents a measurement with values of type string.

Inheritance path: ->Measurement->IdentifiedObject

Table 117 shows all attributes of StringMeasurement.

Table 117 – Attributes of Meas::StringMeasurement

Attribute name	mult.	Attribute type	Description
measurementType	1..1	String	see Measurement
phases	0..1	PhaseCode	see Measurement
unitSymbol	1..1	UnitSymbol	see Measurement
unitMultiplier	1..1	UnitMultiplier	see Measurement
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 118 shows all association ends of Measurement with other classes.

Table 118 – Association ends of Meas::Measurement with other classes

mult.	mult.	Class type name	Description
Terminal (Operation)	0..1	ACDCTerminal	Inherited Association Based On: Meas:: Measurement.Measurements[0..*] --- Core:: ACDCTerminal.Terminal[0..1]
PowerSystemResource (Operation)	1..1	PowerSystemResource	Inherited Association Based On: Core:: PowerSystemResource.PowerSystemResource[0..1] --- Meas:: Measurement.Measurements[0..*]

6.5.26 (Operation) StringMeasurementValue

StringMeasurementValue represents a measurement value of type string.

Inheritance path: ->MeasurementValue->IdentifiedObject

Table 119 shows all attributes of StringMeasurementValue.

Table 119 – Attributes of Meas::StringMeasurementValue

Attribute name	mult.	Attribute type	Description
value (Operation)	1..1	String	The value to supervise.
timeStamp	0..1	DateTime	see MeasurementValue
sensorAccuracy	0..1	PerCent	see MeasurementValue
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 120 shows all association ends of StringMeasurementValue with other classes.

Table 120 – Association ends of Meas::StringMeasurementValue with other classes

mult.	mult.	Class type name	Description
StringMeasurement (Operation)	[1..1]	StringMeasurement	Measurement to which this value is connected. Association Based On: Meas:: StringMeasurementValue.StringMeasurementValues[0..*] --- Meas:: StringMeasurement.StringMeasurement[1]
MeasurementValueSource (Operation)	1..1	MeasurementValueSource	Inherited Association Based On: Meas:: MeasurementValueSource.MeasurementValueSource[1] --- Meas:: MeasurementValue.MeasurementValues[0..*]

6.5.27 (Operation) ValueAliasSet

Describes the translation of a set of values into a name and is intendend to facilitate cusom translations. Each ValueAliasSet has a name, description etc. A specific Measurement may represent a discrete state like Open, Closed, Intermediate etc. This requires a translation from the MeasurementValue.value number to a string, e.g. 0->"Invalid", 1->"Open", 2->"Closed", 3->"Intermediate". Each ValueToAlias member in ValueAliasSet.Value describe a mapping for one particular value to a name.

Inheritance path: ->IdentifiedObject

Table 121 shows all attributes of ValueAliasSet.

Table 121 – Attributes of Meas::ValueAliasSet

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.5.28 (Operation) ValueToAlias

Describes the translation of one particular value into a name, e.g. 1 as "Open".

Inheritance path: ->IdentifiedObject

Table 122 shows all attributes of ValueToAlias.

Table 122 – Attributes of Meas::ValueToAlias

Attribute name	mult.	Attribute type	Description
value (Operation)	1..1	Integer	The value that is mapped.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 123 shows all association ends of ValueToAlias with other classes.

Table 123 – Association ends of Meas::ValueToAlias with other classes

mult.	mult.	Class type name	Description
ValueAliasSet (Operation)	[1..1]	ValueAliasSet	The ValueToAlias mappings included in the set. Association Based On: Meas:: ValueAliasSet.ValueAliasSet[1] ----- Meas:: ValueToAlias.Values[1..*]

6.5.29 Enumerations

6.5.29.1 Source

Source gives information related to the origin of a value.

Literal	Description	Code
PROCESS	The value is provided by input from the process I/O or being calculated from some function.	
DEFAULTED	The value contains a default value.	
SUBSTITUTED	The value is provided by input of an operator or by an automatic source.	

6.5.29.2 Validity

Validity for MeasurementValue.

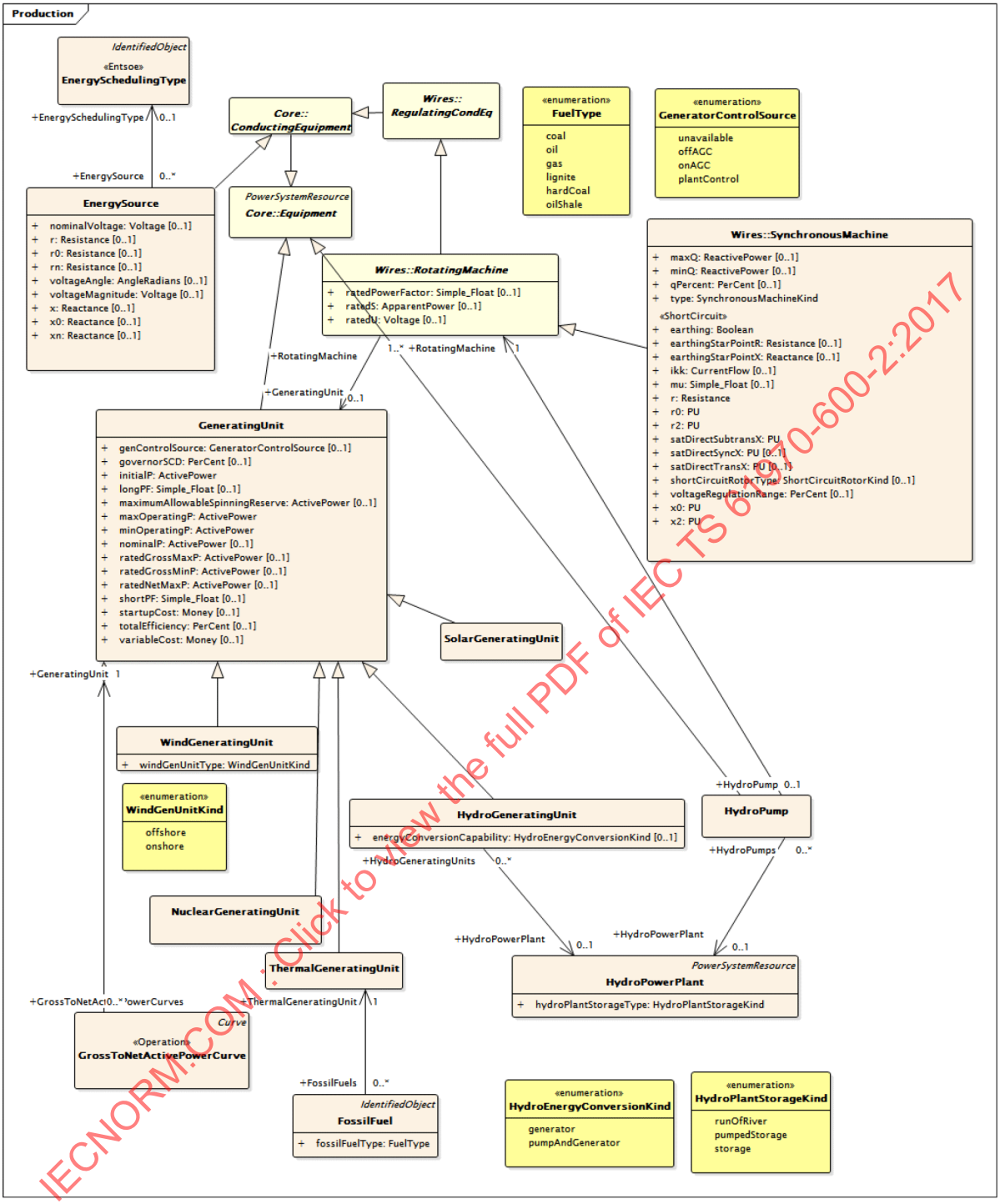
Literal	Description	Code
GOOD	The value is marked good if no abnormal condition of the acquisition function or the information source is detected.	
QUESTIONABLE	The value is marked questionable if a supervision function detects an abnormal behaviour, however the value could still be valid. The client is responsible for determining whether or not values marked "questionable" should be used.	
INVALID	The value is marked invalid when a supervision function recognises abnormal conditions of the acquisition function or the information source (missing or non-operating updating devices). The value is not defined under this condition. The mark invalid is used to indicate to the client that the value may be incorrect and shall not be used.	

6.6 Production

6.6.1 General

The production package is responsible for classes which describe various kinds of generators. These classes also provide production costing information which is used to economically allocate demand among committed units and calculate reserve quantities.

Figure 12 shows the diagram.



IEC

Figure 12 – diagram Production

Figure 13 shows the diagram.

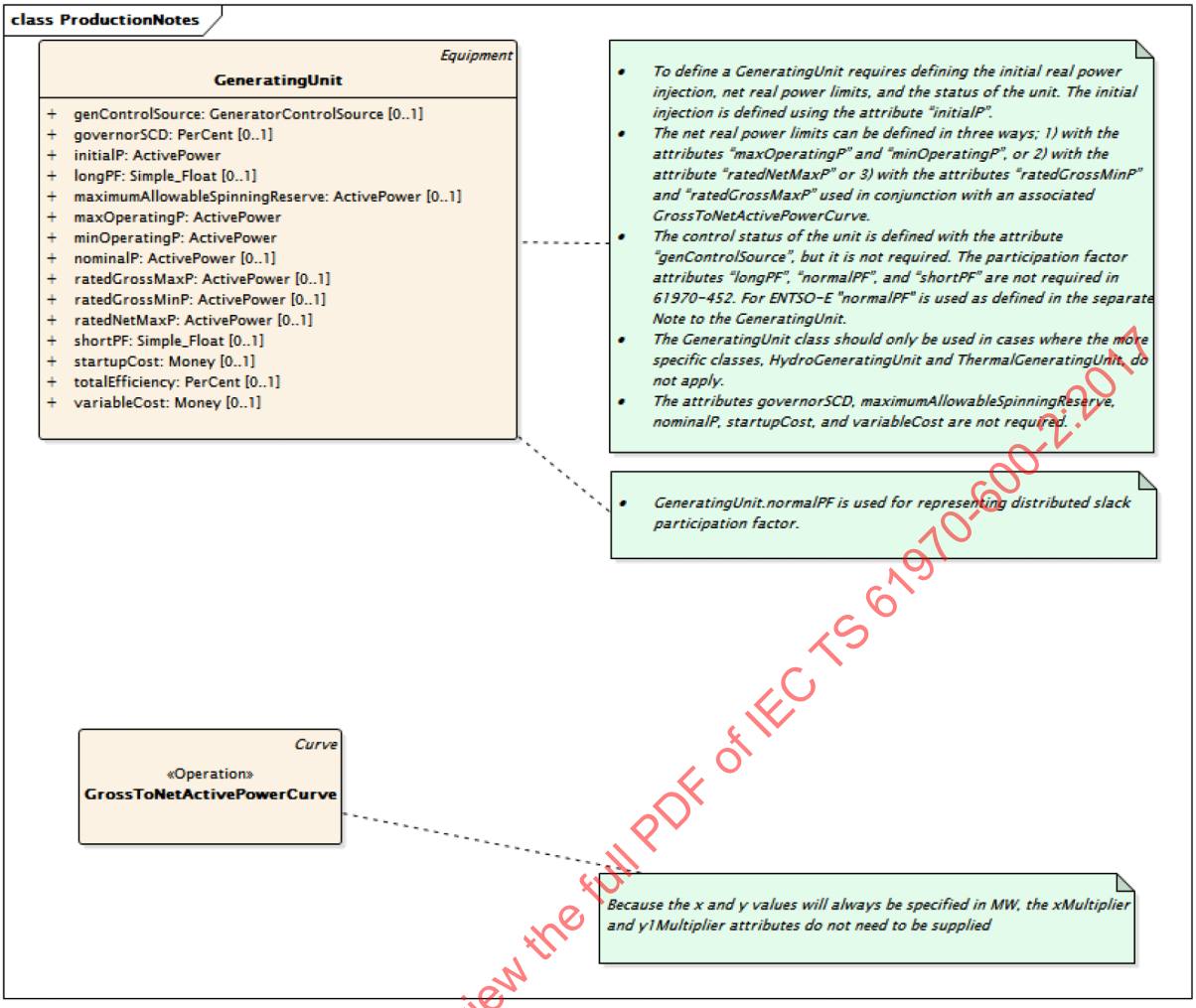


Figure 13 – diagram ProductionNotes

6.6.2 (Entsoe) EnergySchedulingType

Used to define the type of generation for scheduling purposes.

Inheritance path: ->IdentifiedObject

Table 124 shows all attributes of EnergySchedulingType.

Table 124 – Attributes of Production::EnergySchedulingType

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.6.3 EnergySource

A generic equivalent for an energy supplier on a transmission or distribution voltage level.

Inheritance path: ->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 125 shows all attributes of EnergySource.

Table 125 – Attributes of Production::EnergySource

Attribute name	mult.	Attribute type	Description
nominalVoltage	0..1	Voltage	Phase-to-phase nominal voltage.
r	0..1	Resistance	Positive sequence Thevenin resistance.
r0	0..1	Resistance	Zero sequence Thevenin resistance.
rn	0..1	Resistance	Negative sequence Thevenin resistance.
voltageAngle	0..1	AngleRadians	Phase angle of a-phase open circuit.
voltageMagnitude	0..1	Voltage	Phase-to-phase open circuit voltage magnitude.
x	0..1	Reactance	Positive sequence Thevenin reactance.
x0	0..1	Reactance	Zero sequence Thevenin reactance.
xn	0..1	Reactance	Negative sequence Thevenin reactance.
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 126 shows all association ends of EnergySource with other classes.

Table 126 – Association ends of Production::EnergySource with other classes

mult.	mult.	Class type name	Description
EnergySchedulingType	[0..1]	EnergySchedulingType	Energy Source of a particular Energy Scheduling Type Association Based On: Wires:: EnergySource.EnergySource[0..*] ----- Extension:: EnergySchedulingType.EnergySchedulingType[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.6.4 FossilFuel

The fossil fuel consumed by the non-nuclear thermal generating unit. For example, coal, oil, gas, etc. This a the specific fuels that the generating unit can consume.

Inheritance path: ->IdentifiedObject

Table 127 shows all attributes of FossilFuel.

Table 127 – Attributes of Production::FossilFuel

Attribute name	mult.	Attribute type	Description
fossilFuelType	1..1	FuelType	The type of fossil fuel, such as coal, oil, or gas.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 128 shows all association ends of FossilFuel with other classes.

Table 128 – Association ends of Production::FossilFuel with other classes

mult.	mult.	Class type name	Description
ThermalGeneratingUnit	[1..1]	ThermalGeneratingUnit	A thermal generating unit may have one or more fossil fuels. Association Based On: Production:: FossilFuel.FossilFuels[0..*] ----- Production:: ThermalGeneratingUnit.ThermalGeneratingUnit[1]

6.6.5 GeneratingUnit

A single or set of synchronous machines for converting mechanical power into alternating-current power. For example, individual machines within a set may be defined for scheduling purposes while a single control signal is derived for the set. In this case there would be a GeneratingUnit for each member of the set and an additional GeneratingUnit corresponding to the set.

- To define a GeneratingUnit requires defining the initial real power injection, net real power limits, and the status of the unit. The initial injection is defined using the attribute “initialP”.
- The net real power limits can be defined in three ways; 1) with the attributes “maxOperatingP” and “minOperatingP”, or 2) with the attribute “ratedNetMaxP” or 3) with the attributes “ratedGrossMinP” and “ratedGrossMaxP” used in conjunction with an associated GrossToNetActivePowerCurve.
- The control status of the unit is defined with the attribute “genControlSource” but it is not required. The participation factor attributes “longPF”, “normalPF”, and “shortPF” are not required in 61970-452. For ENTSO-E "normalPF" is used as defined in the separate Note to the GeneratingUnit.
- The GeneratingUnit class should only be used in cases where the more specific classes, HydroGeneratingUnit and ThermalGeneratingUnit, do not apply.
- The attributes governorSCD, maximumAllowableSpinningReserve, nominalP, startupCost, and variableCost are not required.
- GeneratingUnit.normalPF is used for representing distributed slack participation factor.

Inheritance path: ->Equipment->PowerSystemResource->IdentifiedObject

Table 129 shows all attributes of GeneratingUnit.

Table 129 – Attributes of Production::GeneratingUnit

Attribute name	mult.	Attribute type	Description
genControlSource	0..1	GeneratorControlSource	The source of controls for a generating unit.
governorSCD	0..1	PerCent	Governor Speed Changer Droop. This is the change in generator power output divided by the change in frequency normalized by the nominal power of the generator and the nominal frequency and expressed in percent and negated. A positive value of speed change droop provides additional generator output upon a drop in frequency.
initialP	1..1	ActivePower	Default initial active power which is used to store a powerflow result for the initial active power for this unit in this network configuration.
longPF	0..1	Simple_Float	Generating unit long term economic participation factor.
maximumAllowableSpinningReserve	0..1	ActivePower	Maximum allowable spinning reserve. Spinning reserve will never be considered greater than this value regardless of the current operating point.
maxOperatingP	1..1	ActivePower	This is the maximum operating active power limit the dispatcher can enter for this unit.
minOperatingP	1..1	ActivePower	This is the minimum operating active power limit the dispatcher can enter for this unit.
nominalP	0..1	ActivePower	The nominal power of the generating unit. Used to give precise meaning to percentage based attributes such as the governor speed change droop (governorSCD attribute). The attribute shall be a positive value equal or less than RotatingMachine.ratedS.
ratedGrossMaxP	0..1	ActivePower	The unit's gross rated maximum capacity (book value).
ratedGrossMinP	0..1	ActivePower	The gross rated minimum generation level which the unit can safely operate at while delivering power to the transmission grid.
ratedNetMaxP	0..1	ActivePower	The net rated maximum capacity determined by subtracting the auxiliary power used to operate the internal plant machinery from the rated gross maximum capacity.
shortPF	0..1	Simple_Float	Generating unit short term economic participation factor.
startupCost	0..1	Money	The initial startup cost incurred for each start of the GeneratingUnit.
variableCost	0..1	Money	The variable cost component of production per unit of ActivePower.
totalEfficiency	0..1	PerCent	The efficiency of the unit in converting the fuel into electrical energy.
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject

Attribute name	mult.	Attribute type	Description
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 130 shows all association ends of Equipment with other classes.

Table 130 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	0..1	`EquipmentContainer`	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.6.6 (Operation) GrossToNetActivePowerCurve

Relationship between the generating unit's gross active power output on the X-axis (measured at the terminals of the machine(s)) and the generating unit's net active power output on the Y-axis (based on utility-defined measurements at the power station). Station service loads, when modeled, should be treated as non-conforming bus loads. There may be more than one curve, depending on the auxiliary equipment that is in service.

Because the x and y values will always be specified in MW, the xMultiplier and y1Multiplier attributes do not need to be supplied

Inheritance path: ->Curve->IdentifiedObject

Table 131 shows all attributes of GrossToNetActivePowerCurve.

Table 131 – Attributes of Production::GrossToNetActivePowerCurve

Attribute name	mult.	Attribute type	Description
curveStyle	1..1	CurveStyle	see Curve
xUnit	1..1	UnitSymbol	see Curve
y1Unit	1..1	UnitSymbol	see Curve
y2Unit	0..1	UnitSymbol	see Curve
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 132 shows all association ends of GrossToNetActivePowerCurve with other classes.

Table 132 – Association ends of Production::GrossToNetActivePowerCurve with other classes

mult.	mult.	Class type name	Description
GeneratingUnit (Operation)	[1..1]	GeneratingUnit	A generating unit may have a gross active power to net active power curve, describing the losses and auxiliary power requirements of the unit. Association Based On: Production:: GrossToNetActivePowerCurve.GrossToNetActivePowerCurves[0..*] ---- - Production:: GeneratingUnit.GeneratingUnit[1]

6.6.7 HydroGeneratingUnit

A generating unit whose prime mover is a hydraulic turbine (e.g., Francis, Pelton, Kaplan).

Inheritance path: ->GeneratingUnit->Equipment->PowerSystemResource->IdentifiedObject

Table 133 shows all attributes of HydroGeneratingUnit.

Table 133 – Attributes of Production::HydroGeneratingUnit

Attribute name	mult.	Attribute type	Description
energyConversionCapability	0..1	HydroEnergyConversionKind	Energy conversion capability for generating.
genControlSource	0..1	GeneratorControlSource	see GeneratingUnit
governorSCD	0..1	PerCent	see GeneratingUnit
initialP	1..1	ActivePower	see GeneratingUnit
longPF	0..1	Simple_Float	see GeneratingUnit
maximumAllowableSpinningReserve	0..1	ActivePower	see GeneratingUnit
maxOperatingP	1..1	ActivePower	see GeneratingUnit
minOperatingP	1..1	ActivePower	see GeneratingUnit
nominalP	0..1	ActivePower	see GeneratingUnit
ratedGrossMaxP	0..1	ActivePower	see GeneratingUnit
ratedGrossMinP	0..1	ActivePower	see GeneratingUnit
ratedNetMaxP	0..1	ActivePower	see GeneratingUnit
shortPF	0..1	Simple_Float	see GeneratingUnit
startupCost	0..1	Money	see GeneratingUnit
variableCost	0..1	Money	see GeneratingUnit
totalEfficiency	0..1	PerCent	see GeneratingUnit
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 134 shows all association ends of HydroGeneratingUnit with other classes.

Table 134 – Association ends of Production::HydroGeneratingUnit with other classes

mult.	mult.	Class type name	Description
HydroPowerPlant	[0..1]	HydroPowerPlant	The hydro generating unit belongs to a hydro power plant. Association Based On: Production:: HydroPowerPlant.HydroPowerPlant[0..1] ----- Production:: HydroGeneratingUnit.HydroGeneratingUnits[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.6.8 HydroPowerPlant

A hydro power station which can generate or pump. When generating, the generator turbines receive water from an upper reservoir. When pumping, the pumps receive their water from a lower reservoir.

Inheritance path: ->PowerSystemResource->IdentifiedObject

Table 135 shows all attributes of HydroPowerPlant.

Table 135 – Attributes of Production::HydroPowerPlant

Attribute name	mult.	Attribute type	Description
hydroPlantStorageType	1..1	HydroPlantStorageKind	The type of hydro power plant water storage.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.6.9 HydroPump

A synchronous motor-driven pump, typically associated with a pumped storage plant.

Inheritance path: ->Equipment->PowerSystemResource->IdentifiedObject

Table 136 shows all attributes of HydroPump.

Table 136 – Attributes of Production::HydroPump

Attribute name	mult.	Attribute type	Description
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 137 shows all association ends of HydroPump with other classes.

Table 137 – Association ends of Production::HydroPump with other classes

mult.	mult.	Class type name	Description
RotatingMachine	[1..1]	RotatingMachine	The synchronous machine drives the turbine which moves the water from a low elevation to a higher elevation. The direction of machine rotation for pumping may or may not be the same as for generating. Association Based On: Wires:: RotatingMachine.RotatingMachine[1] ----- Production:: HydroPump.HydroPump[0..1]
HydroPowerPlant	[0..1]	HydroPowerPlant	The hydro pump may be a member of a pumped storage plant or a pump for distributing water. Association Based On: Production:: HydroPowerPlant.HydroPowerPlant[0..1] ----- Production:: HydroPump.HydroPumps[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.6.10 NuclearGeneratingUnit

A nuclear generating unit.

Inheritance path: ->GeneratingUnit->Equipment->PowerSystemResource->IdentifiedObject

Table 138 shows all attributes of NuclearGeneratingUnit.

Table 138 – Attributes of Production::NuclearGeneratingUnit

Attribute name	mult.	Attribute type	Description
genControlSource	0..1	GeneratorControlSource	see GeneratingUnit
governorSCD	0..1	PerCent	see GeneratingUnit
initialP	1..1	ActivePower	see GeneratingUnit
longPF	0..1	Simple_Float	see GeneratingUnit
maximumAllowableSpinningReserve	0..1	ActivePower	see GeneratingUnit
maxOperatingP	1..1	ActivePower	see GeneratingUnit
minOperatingP	1..1	ActivePower	see GeneratingUnit
nominalP	0..1	ActivePower	see GeneratingUnit
ratedGrossMaxP	0..1	ActivePower	see GeneratingUnit
ratedGrossMinP	0..1	ActivePower	see GeneratingUnit
ratedNetMaxP	0..1	ActivePower	see GeneratingUnit
shortPF	0..1	Simple_Float	see GeneratingUnit
startupCost	0..1	Money	see GeneratingUnit
variableCost	0..1	Money	see GeneratingUnit
totalEfficiency	0..1	PerCent	see GeneratingUnit
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 139 shows all association ends of Equipment with other classes.

Table 139 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] --- Core:: Equipment.Equipments[0..*]

6.6.11 SolarGeneratingUnit

A solar thermal generating unit.

Inheritance path: ->GeneratingUnit->Equipment->PowerSystemResource->IdentifiedObject

Table 140 shows all attributes of SolarGeneratingUnit.

Table 140 – Attributes of Production::SolarGeneratingUnit

Attribute name	mult.	Attribute type	Description
genControlSource	0..1	GeneratorControlSource	see GeneratingUnit
governorSCD	0..1	PerCent	see GeneratingUnit
initialP	1..1	ActivePower	see GeneratingUnit
longPF	0..1	Simple_Float	see GeneratingUnit
maximumAllowableSpinningReserve	0..1	ActivePower	see GeneratingUnit
maxOperatingP	1..1	ActivePower	see GeneratingUnit
minOperatingP	1..1	ActivePower	see GeneratingUnit
nominalP	0..1	ActivePower	see GeneratingUnit
ratedGrossMaxP	0..1	ActivePower	see GeneratingUnit
ratedGrossMinP	0..1	ActivePower	see GeneratingUnit
ratedNetMaxP	0..1	ActivePower	see GeneratingUnit
shortPF	0..1	Simple_Float	see GeneratingUnit
startupCost	0..1	Money	see GeneratingUnit
variableCost	0..1	Money	see GeneratingUnit
totalEfficiency	0..1	PerCent	see GeneratingUnit
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 141 shows all association ends of Equipment with other classes.

Table 141 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.6.12 ThermalGeneratingUnit

A generating unit whose prime mover could be a steam turbine, combustion turbine, or diesel engine.

Inheritance path: ->GeneratingUnit->Equipment->PowerSystemResource->IdentifiedObject

Table 142 shows all attributes of ThermalGeneratingUnit.

Table 142 – Attributes of Production::ThermalGeneratingUnit

Attribute name	mult.	Attribute type	Description
genControlSource	0..1	GeneratorControlSource	see GeneratingUnit
governorSCD	0..1	PerCent	see GeneratingUnit
initialP	1..1	ActivePower	see GeneratingUnit
longPF	0..1	Simple_Float	see GeneratingUnit
maximumAllowableSpinningReserve	0..1	ActivePower	see GeneratingUnit
maxOperatingP	1..1	ActivePower	see GeneratingUnit
minOperatingP	1..1	ActivePower	see GeneratingUnit
nominalP	0..1	ActivePower	see GeneratingUnit
ratedGrossMaxP	0..1	ActivePower	see GeneratingUnit
ratedGrossMinP	0..1	ActivePower	see GeneratingUnit
ratedNetMaxP	0..1	ActivePower	see GeneratingUnit
shortPF	0..1	Simple_Float	see GeneratingUnit
startupCost	0..1	Money	see GeneratingUnit
variableCost	0..1	Money	see GeneratingUnit
totalEfficiency	0..1	PerCent	see GeneratingUnit
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 143 shows all association ends of Equipment with other classes.

Table 143 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.6.13 WindGeneratingUnit

A wind driven generating unit. May be used to represent a single turbine or an aggregation.

Inheritance path: ->GeneratingUnit->Equipment->PowerSystemResource->IdentifiedObject

Table 144 shows all attributes of WindGeneratingUnit.

Table 144 – Attributes of Production::WindGeneratingUnit

Attribute name	mult.	Attribute type	Description
windGenUnitType	1..1	WindGenUnitKind	The kind of wind generating unit
genControlSource	0..1	GeneratorControlSource	see GeneratingUnit
governorSCD	0..1	PerCent	see GeneratingUnit
initialP	1..1	ActivePower	see GeneratingUnit
longPF	0..1	Simple_Float	see GeneratingUnit
maximumAllowableSpinningReserve	0..1	ActivePower	see GeneratingUnit
maxOperatingP	1..1	ActivePower	see GeneratingUnit
minOperatingP	1..1	ActivePower	see GeneratingUnit
nominalP	0..1	ActivePower	see GeneratingUnit
ratedGrossMaxP	0..1	ActivePower	see GeneratingUnit
ratedGrossMinP	0..1	ActivePower	see GeneratingUnit
ratedNetMaxP	0..1	ActivePower	see GeneratingUnit
shortPF	0..1	Simple_Float	see GeneratingUnit
startupCost	0..1	Money	see GeneratingUnit
variableCost	0..1	Money	see GeneratingUnit
totalEfficiency	0..1	PerCent	see GeneratingUnit
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 145 shows all association ends of Equipment with other classes.

Table 145 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] --- Core:: Equipment.Equipments[0..*]

6.6.14 Enumerations

6.6.14.1 FuelType

Type of fuel.

Literal	Description	Code
coal	Generic coal, not including lignite type.	
oil	Oil.	
gas	Natural gas.	
lignite	The fuel is lignite coal. Note that this is a special type of coal, so the other enum of coal is reserved for hard coal types or if the exact type of coal is not known.	
hardCoal	Hard coal	
oilShale	Oil Shale	

6.6.14.2 GeneratorControlSource

The source of controls for a generating unit.

Literal	Description	Code
unavailable	Not available.	
offAGC	Off of automatic generation control (AGC).	
onAGC	On automatic generation control (AGC).	
plantControl	Plant is controlling.	

6.6.14.3 HydroEnergyConversionKind

Specifies the capability of the hydro generating unit to convert energy as a generator or pump.

Literal	Description	Code
generator	Able to generate power, but not able to pump water for energy storage.	
pumpAndGenerator	Able to both generate power and pump water for energy storage.	

6.6.14.4 HydroPlantStorageKind

The type of hydro power plant.

Literal	Description	Code
runOfRiver	Run of river.	
pumpedStorage	Pumped storage.	
storage	Storage.	

6.6.14.5 WindGenUnitKind

Kind of wind generating unit.

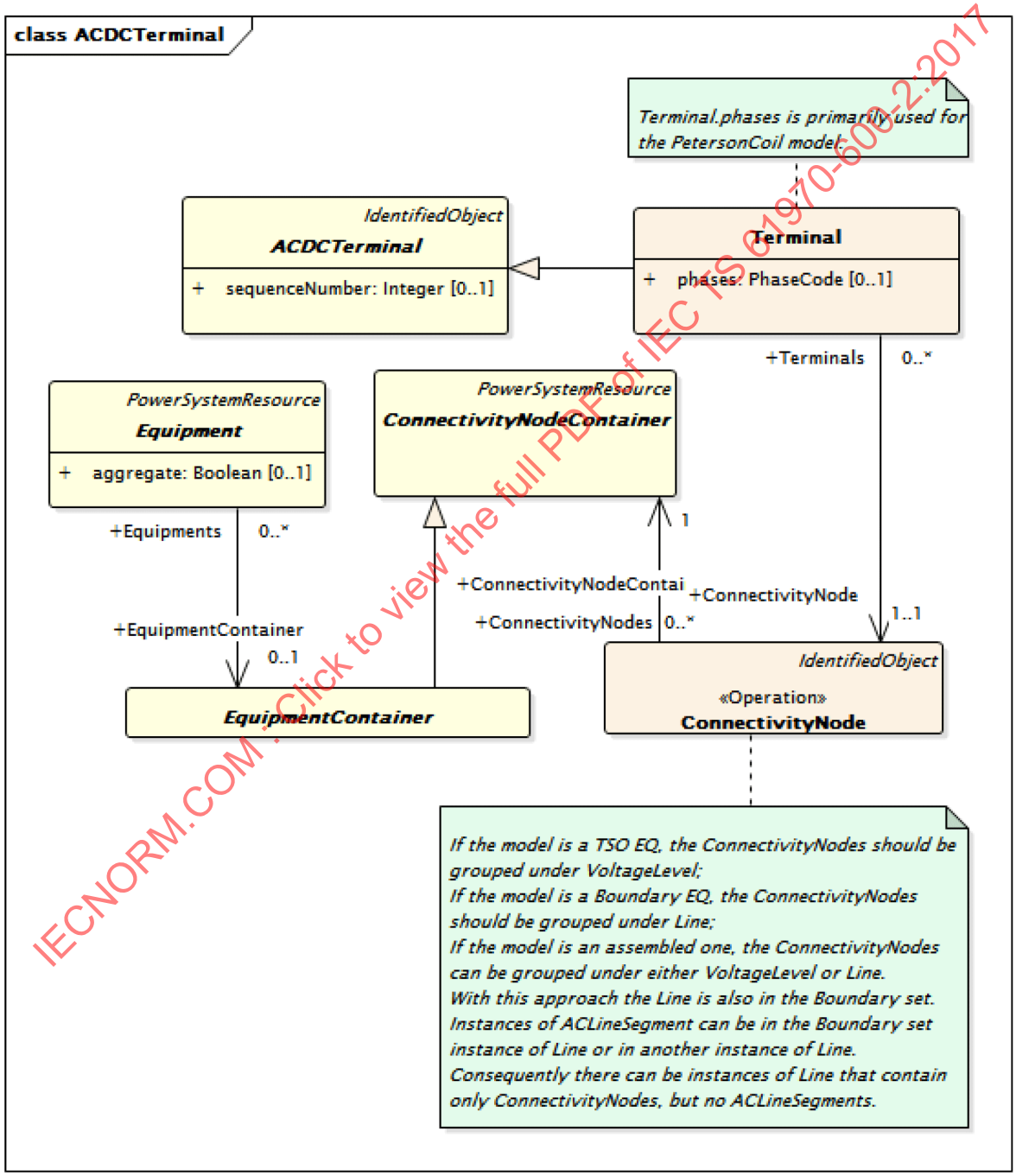
Literal	Description	Code
offshore	The wind generating unit is located offshore.	
onshore	The wind generating unit is located onshore.	

6.7 Core

6.7.1 General

Contains the core PowerSystemResource and ConductingEquipment entities shared by all applications plus common collections of those entities. Not all applications require all the Core entities. This package does not depend on any other package except the Domain package, but most of the other packages have associations and generalizations that depend on it.

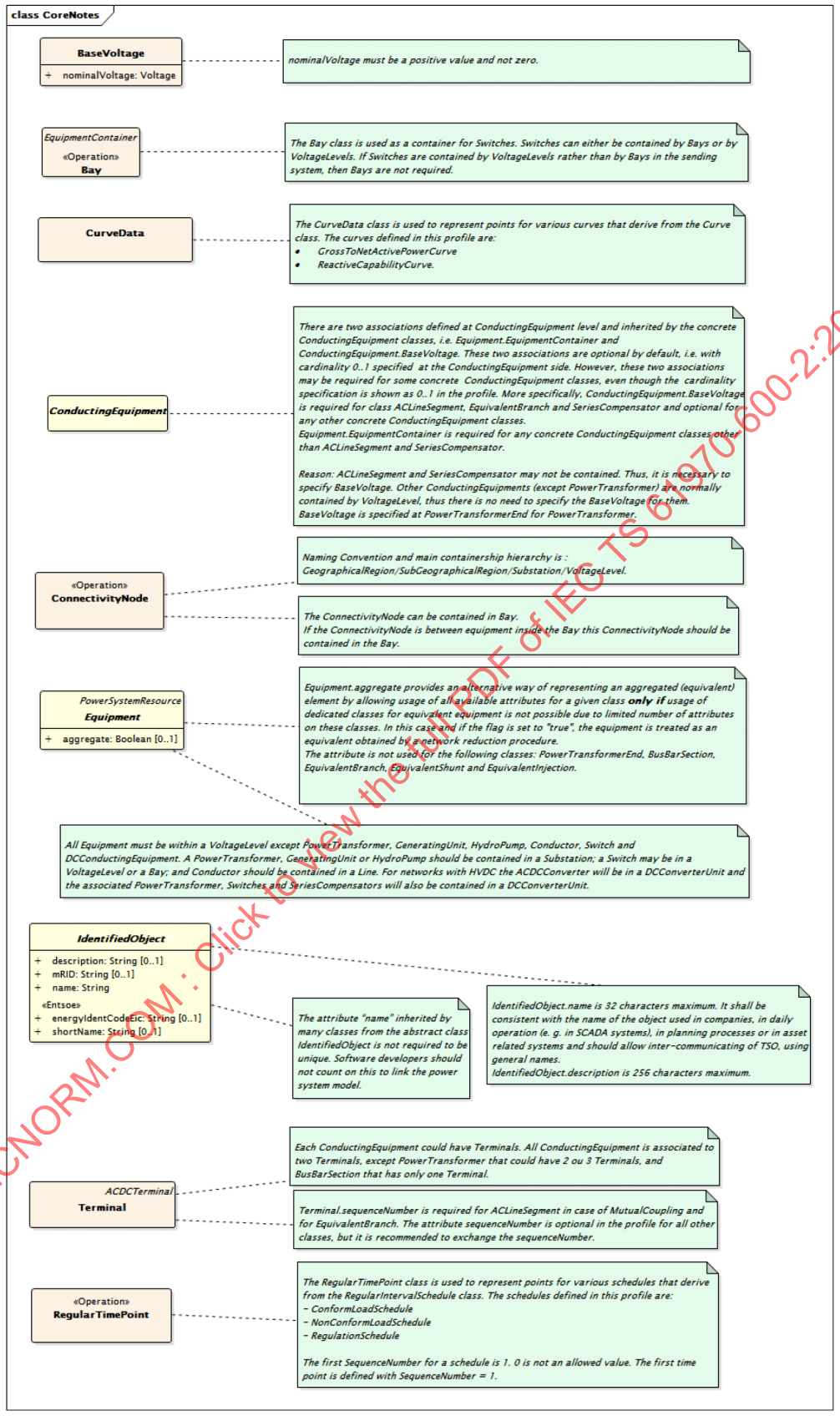
Figure 14 shows the diagram.



IEC

Figure 14 – diagram ACDCTerminal

Figure 15 shows the diagram.



IEC

Figure 15 – diagram CoreNotes

6.7.2 ACDCTerminal(Abstract)

An electrical connection point (AC or DC) to a piece of conducting equipment. Terminals are connected at physical connection points called connectivity nodes.

Inheritance path: ->IdentifiedObject

Table 146 shows all attributes of ACDCTerminal.

Table 146 – Attributes of Core::ACDCTerminal

Attribute name	mult.	Attribute type	Description
sequenceNumber	0..1	Integer	The orientation of the terminal connections for a multiple terminal conducting equipment. The sequence numbering starts with 1 and additional terminals should follow in increasing order. The first terminal is the "starting point" for a two terminal branch.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 147 shows all association ends of ACDCTerminal with other classes.

Table 147 – Association ends of Core::ACDCTerminal with other classes

mult.	mult.	Class type name	Description
BusNameMarker	[0..1]	BusNameMarker	The bus name marker used to name the bus (topological node). Association Based On: Topology:: BusNameMarker.BusNameMarker[0..1] ----- Core:: ACDCTerminal.Terminal[1..*]

6.7.3 BaseVoltage

Defines a system base voltage which is referenced.

OCL constraint:NominalVoltage must be positive

nominalVoltage must be a positive value and not zero.

Inheritance path: ->IdentifiedObject

Table 148 shows all attributes of BaseVoltage.

Table 148 – Attributes of Core::BaseVoltage

Attribute name	mult.	Attribute type	Description
nominalVoltage	1..1	Voltage	The power system resource's base voltage.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.7.4 BasicIntervalSchedule(Abstract)

Schedule of values at points in time.

Inheritance path: ->IdentifiedObject

Table 149 shows all attributes of BasicIntervalSchedule.

Table 149 – Attributes of Core::BasicIntervalSchedule

Attribute name	mult.	Attribute type	Description
startTime	1..1	DateTime	The time for the first time point.
value1Unit	1..1	UnitSymbol	Value1 units of measure.
value2Unit	0..1	UnitSymbol	Value2 units of measure.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.7.5 (Operation) Bay

A collection of power system resources (within a given substation) including conducting equipment, protection relays, measurements, and telemetry. A bay typically represents a physical grouping related to modularization of equipment.

The Bay class is used as a container for Switches. Switches can either be contained by Bays or by VoltageLevels. If Switches are contained by VoltageLevels rather than by Bays in the sending system, then Bays are not required.

Inheritaance path: ->EquipmentContainer->ConnectivityNodeContainer->PowerSystemResource->IdentifiedObject

Table 150 shows all attributes of Bay.

Table 150 – Attributes of Core::Bay

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 151 shows all association ends of Bay with other classes.

Table 151 – Association ends of Core::Bay with other classes

mult.	mult.	Class type name	Description
VoltageLevel (Operation)	[1..1]	VoltageLevel	The voltage level containing this bay. Association Based On: Core:: VoltageLevel.VoltageLevel[0..1] ----- Core:: Bay.Bays[0..*]

6.7.6 ConductingEquipment(Abstract)

The parts of the AC power system that are designed to carry current or that are conductively connected through terminals.

OCL constraint:The ConductingEquipment must either have a BaseVoltage association or be within a VoltageLevel. If both are set the VoltageLevel's BaseVoltage must equal that of the ConductingEquipment

There are two associations defined at ConductingEquipment level and inherited by the concrete ConductingEquipment classes, i.e. Equipment.EquipmentContainer and ConductingEquipment.BaseVoltage. These two associations are optional by default, i.e. with cardinality 0..1 specified at the ConductingEquipment side. However, these two associations may be required for some concrete ConductingEquipment classes, even though the cardinality specification is shown as 0..1 in the profile. More specifically, ConductingEquipment.BaseVoltage is required for class ACLineSegment, EquivalentBranch and SeriesCompensator and optional for any other concrete ConductingEquipment classes. Equipment.EquipmentContainer is required for any concrete ConductingEquipment classes other than ACLineSegment and SeriesCompensator. Reason: ACLineSegment and SeriesCompensator may not be contained. Thus, it is necessary to specify BaseVoltage. Other ConductingEquipments (except PowerTransformer) are normally contained by VoltageLevel, thus there is no need to specify the BaseVoltage for them. BaseVoltage is specified at PowerTransformerEnd for PowerTransformer.

Inheritance path: ->Equipment->PowerSystemResource->IdentifiedObject

Table 152 shows all attributes of ConductingEquipment.

Table 152 – Attributes of Core::ConductingEquipment

Attribute name	mult.	Attribute type	Description
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 153 shows all association ends of ConductingEquipment with other classes.

Table 153 – Association ends of Core::ConductingEquipment with other classes

mult.	mult.	Class type name	Description
BaseVoltage	[0..1]	BaseVoltage	All conducting equipment with this base voltage. Use only when there is no voltage level container used and only one base voltage applies. For example, not used for transformers. Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.7.7 (Operation) ConnectivityNode

Connectivity nodes are points where terminals of AC conducting equipment are connected together with zero impedance.

If the model is a TSO EQ, the ConnectivityNodes should be grouped under VoltageLevel; If the model is a Boundary EQ, the ConnectivityNodes should be grouped under Line; If the model is an assembled one, the ConnectivityNodes can be grouped under either VoltageLevel or Line. With this approach the Line is also in the Boundary set. Instances of ACLineSegment can be in the Boundary set instance of Line or in another instance of Line. Consequently there can be instances of Line that contain only ConnectivityNodes, but no ACLineSegments.

Naming Convention and main containership hierarchy is:
GeographicalRegion/SubGeographicalRegion/Substation/VoltageLevel.

The ConnectivityNode can be contained in Bay. If the ConnectivityNode is between equipment inside the Bay this ConnectivityNode should be contained in the Bay.

Inheritance path: ->IdentifiedObject

Table 154 shows all attributes of ConnectivityNode.

Table 154 – Attributes of Core::ConnectivityNode

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 155 shows all association ends of ConnectivityNode with other classes.

Table 155 – Association ends of Core::ConnectivityNode with other classes

mult.	mult.	Class type name	Description
ConnectivityNodeContainer (Operation)	[1..1]	ConnectivityNodeContainer	Container of this connectivity node. Association Based On: Core:: ConnectivityNode.ConnectivityNodes[0..*] ----- Core:: ConnectivityNodeContainer.ConnectivityNodeContainer[1]

6.7.8 ConnectivityNodeContainer(Abstract)

A base class for all objects that may contain connectivity nodes or topological nodes.

Inheritance path: ->PowerSystemResource->IdentifiedObject

Table 156 shows all attributes of ConnectivityNodeContainer.

Table 156 – Attributes of Core::ConnectivityNodeContainer

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.7.9 Curve(Abstract)

A multi-purpose curve or functional relationship between an independent variable (X-axis) and dependent (Y-axis) variables.

Inheritance path: ->IdentifiedObject

Table 157 shows all attributes of Curve.

Table 157 – Attributes of Core::Curve

Attribute name	mult.	Attribute type	Description
curveStyle	1..1	CurveStyle	The style or shape of the curve.
xUnit	1..1	UnitSymbol	The X-axis units of measure.
y1Unit	1..1	UnitSymbol	The Y1-axis units of measure.
y2Unit	0..1	UnitSymbol	The Y2-axis units of measure.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.7.10 CurveData

Multi-purpose data points for defining a curve. The use of this generic class is discouraged if a more specific class can be used to specify the x and y axis values along with their specific data types.

The CurveData class is used to represent points for various curves that derive from the Curve class. The curves defined in this profile are:

- GrossToNetActivePowerCurve
- ReactiveCapabilityCurve.

Table 158 shows all attributes of CurveData.

Table 158 – Attributes of Core::CurveData

Attribute name	mult.	Attribute type	Description
xvalue	1..1	Simple_Float	The data value of the X-axis variable, depending on the X-axis units.
y1value	1..1	Simple_Float	The data value of the first Y-axis variable, depending on the Y-axis units.
y2value	0..1	Simple_Float	The data value of the second Y-axis variable (if present), depending on the Y-axis units.

Table 159 shows all association ends of CurveData with other classes.

Table 159 – Association ends of Core::CurveData with other classes

mult.	mult.	Class type name	Description
Curve	[1..1]	Curve	The point data values that define this curve. Association Based On: Core:: CurveData.CurveDatas[0..*] ----- Core:: Curve.Curve[1]

6.7.11 Equipment(Abstract)

The parts of a power system that are physical devices, electronic or mechanical.

OCL constraint:OCL on containment. Please refer to the Note linked to the Equipment class

Equipment.aggregate provides an alternative way of representing an aggregated (equivalent) element by allowing usage of all available attributes for a given class **only if** usage of dedicated classes for equivalent equipment is not possible due to limited number of attributes on these classes. In this case and if the flag is set to "true", the equipment is treated as an equivalent obtained by a network reduction procedure. The attribute is not used for the following classes: PowerTransformerEnd, BusBarSection, EquivalentBranch, EquivalentShunt and EquivalentInjection.

All Equipment must be within a VoltageLevel except PowerTransformer, GeneratingUnit, HydroPump, Conductor, Switch and DCConductingEquipment. A PowerTransformer, GeneratingUnit or HydroPump should be contained in a Substation; a Switch may be in a VoltageLevel or a Bay; and Conductor should be contained in a Line. For networks with HVDC the ACDCConverter will be in a DCConverterUnit and the associated PowerTransformer, Switches and SeriesCompensators will also be contained in a DCConverterUnit.

Inheritance path: ->PowerSystemResource->IdentifiedObject

Table 160 shows all attributes of Equipment.

Table 160 – Attributes of Core::Equipment

Attribute name	mult.	Attribute type	Description
aggregate	0..1	Boolean	The single instance of equipment represents multiple pieces of equipment that have been modeled together as an aggregate. Examples would be power transformers or synchronous machines operating in parallel modeled as a single aggregate power transformer or aggregate synchronous machine. This is not to be used to indicate equipment that is part of a group of interdependent equipment produced by a network production program.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 161 shows all association ends of Equipment with other classes.

Table 161 – Association ends of Core::Equipment with other classes

mult.	mult.	Class type name	Description
EquipmentContainer	[0..1]	EquipmentContainer	Container of this equipment. Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.7.12 EquipmentContainer(Abstract)

A modeling construct to provide a root class for containing equipment.

Inheritance path: ->ConnectivityNodeContainer->PowerSystemResource->IdentifiedObject

Table 162 shows all attributes of EquipmentContainer.

Table 162 – Attributes of Core::EquipmentContainer

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.7.13 GeographicalRegion

A geographical region of a power system network model.

Inheritance path: ->IdentifiedObject

Table 163 shows all attributes of GeographicalRegion.

Table 163 – Attributes of Core::GeographicalRegion

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.7.14 IdentifiedObject(Abstract)

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

OCL constraint: R.4.10.11. Description length restriction (optional)

OCL constraint: R.4.10.11. Energy Ident Code length restriction (optional)

OCL constraint: R.4.10.11. Name length restriction

OCL constraint: R.4.10.11. ShortName length restriction (optional)

IdentifiedObject.name is 32 characters maximum. It shall be consistent with the name of the object used in companies, in daily operation (e. g. in SCADA systems), in planning processes or in asset related systems and should allow inter-communicating of TSO, using general names. IdentifiedObject.description is 256 characters maximum.

The attribute “name” inherited by many classes from the abstract class IdentifiedObject is not required to be unique. Software developers should not count on this to link the power system model.

Table 164 shows all attributes of IdentifiedObject.

Table 164 – Attributes of Core::IdentifiedObject

Attribute name	mult.	Attribute type	Description
description	0..1	String	The description is a free human readable text describing or naming the object. It may be non unique and may not correlate to a naming hierarchy.
energyIdentCodeEic (Entsoe)	0..1	String	The attribute is used for an exchange of the EIC code (Energy identification Code). The length of the string is 16 characters as defined by the EIC code. References: Local issuing offices for EIC: https://www.entsoe.eu/publications/edi-library/links-to-eic-websites/ EIC description: https://www.entsoe.eu/index.php?id=73&libCat=eic .
mRID	0..1	String	Master resource identifier issued by a model authority. The mRID is globally 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 61970-552:2013, the mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements.
name	1..1	String	The name is any free human readable and possibly non unique text naming the object.
shortName (Entsoe)	0..1	String	The attribute is used for an exchange of a human readable short name with length of the string 12 characters maximum.

6.7.15 PowerSystemResource(Abstract)

A power system resource can be an item of equipment such as a switch, an equipment container containing many individual items of equipment such as a substation, or an organisational entity such as sub-control area. Power system resources can have measurements associated.

Inheritance path: ->IdentifiedObject

Table 165 shows all attributes of PowerSystemResource.

Table 165 – Attributes of Core::PowerSystemResource

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.7.16 RegularIntervalSchedule(Abstract)

The schedule has time points where the time between them is constant.

Inheritance path: ->BasicIntervalSchedule->IdentifiedObject

Table 166 shows all attributes of RegularIntervalSchedule.

Table 166 – Attributes of Core::RegularIntervalSchedule

Attribute name	mult.	Attribute type	Description
timeStep	1..1	Seconds	The time between each pair of subsequent regular time points in sequence order.
endTime	1..1	DateTime	The time for the last time point.
startTime	1..1	DateTime	see BasicIntervalSchedule
value1Unit	1..1	UnitSymbol	see BasicIntervalSchedule
value2Unit	0..1	UnitSymbol	see BasicIntervalSchedule
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.7.17 (Operation) RegularTimePoint

Time point for a schedule where the time between the consecutive points is constant.

The RegularTimePoint class is used to represent points for various schedules that derive from the RegularIntervalSchedule class. The schedules defined in this profile are: – ConformLoadSchedule – NonConformLoadSchedule – RegulationSchedule The first SequenceNumber for a schedule is 1. 0 is not an allowed value. The first time point is defined with SequenceNumber = 1.

Table 167 shows all attributes of RegularTimePoint.

Table 167 – Attributes of Core::RegularTimePoint

Attribute name	mult.	Attribute type	Description
sequenceNumber (Operation)	1..1	Integer	The position of the regular time point in the sequence. Note that time points don't have to be sequential, i.e. time points may be omitted. The actual time for a RegularTimePoint is computed by multiplying the associated regular interval schedule's time step with the regular time point sequence number and adding the associated schedules start time.
value1 (Operation)	1..1	Simple_Float	The first value at the time. The meaning of the value is defined by the derived type of the associated schedule.
value2 (Operation)	0..1	Simple_Float	The second value at the time. The meaning of the value is defined by the derived type of the associated schedule.

Table 168 shows all association ends of RegularTimePoint with other classes.

Table 168 – Association ends of Core::RegularTimePoint with other classes

mult.	mult.	Class type name	Description
IntervalSchedule (Operation)	[1..1]	RegularIntervalSchedule	Regular interval schedule containing this time point. Association Based On: Core:: RegularIntervalSchedule.IntervalSchedule[1] ----- Core:: RegularTimePoint.TimePoints[1..*]

6.7.18 ReportingGroup

A reporting group is used for various ad-hoc groupings used for reporting.

Inheritance path: ->IdentifiedObject

Table 169 shows all attributes of ReportingGroup.

Table 169 – Attributes of Core::ReportingGroup

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.7.19 SubGeographicalRegion

A subset of a geographical region of a power system network model.

Inheritance path: ->IdentifiedObject

Table 170 shows all attributes of SubGeographicalRegion.

Table 170 – Attributes of Core::SubGeographicalRegion

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 171 shows all association ends of SubGeographicalRegion with other classes.

Table 171 – Association ends of Core::SubGeographicalRegion with other classes

mult.	mult.	Class type name	Description
Region	[1..1]	GeographicalRegion	The geographical region to which this sub-geographical region is within. Association Based On: Core:: GeographicalRegion.Region[0..1] ----- Core:: SubGeographicalRegion.Regions[0..*]

6.7.20 Substation

A collection of equipment for purposes other than generation or utilization, through which electric energy in bulk is passed for the purposes of switching or modifying its characteristics.

Inheritance path: ->EquipmentContainer->ConnectivityNodeContainer->PowerSystemResource->IdentifiedObject

Table 172 shows all attributes of Substation.

Table 172 – Attributes of Core::Substation

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 173 shows all association ends of Substation with other classes.

Table 173 – Association ends of Core::Substation with other classes

mult.	mult.	Class type name	Description
Region	[1..1]	SubGeographicalRegion	The SubGeographicalRegion containing the substation. Association Based On: Core:: SubGeographicalRegion.Region[0..1] ----- Core:: Substation.Substations[0..*]

6.7.21 Terminal

An AC electrical connection point to a piece of conducting equipment. Terminals are connected at physical connection points called connectivity nodes.

OCL constraint:Sequence Number is required for EquivalentBranch and ACLineSegments with MutualCoupling

Terminal.phases is primarily used for the PetersonCoil model.

Terminal.sequenceNumber is required for ACLineSegment in case of MutualCoupling and for EquivalentBranch. The attribute sequenceNumber is optional in the profile for all other classes, but it is recommended to exchange the sequenceNumber.

Each ConductingEquipment could have Terminals. All ConductingEquipment is associated to two Terminals, except PowerTransformer that could have 2 ou 3 Terminals, and BusBarSection that has only one Terminal.

Inheritance path: ->ACDCTerminal->IdentifiedObject

Table 174 shows all attributes of Terminal.

Table 174 – Attributes of Core::Terminal

Attribute name	mult.	Attribute type	Description
phases	0..1	PhaseCode	Represents the normal network phasing condition. If the attribute is missing three phases (ABC or ABCN) shall be assumed.
sequenceNumber	0..1	Integer	see ACDCTerminal
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 175 shows all association ends of Terminal with other classes.

Table 175 – Association ends of Core::Terminal with other classes

mult.	mult.	Class type name	Description
ConnectivityNode	[1..1]	ConnectivityNode	Terminals interconnected with zero impedance at a this connectivity node. Association Based On: Core:: Terminal.Terminals[0..*] ----- Core:: ConnectivityNode.ConnectivityNode[0..1]
ConductingEquipment	[1..1]	ConductingEquipment	The conducting equipment of the terminal. Conducting equipment have terminals that may be connected to other conducting equipment terminals via connectivity nodes or topological nodes. Association Based On: Core:: Terminal.Terminals[0..*] ----- Core:: ConductingEquipment.ConductingEquipment[1]
BusNameMarker	0..1	BusNameMarker	Inherited Association Based On: Topology:: BusNameMarker.BusNameMarker[0..1] ----- Core:: ACDCTerminal.Terminal[1..*]

6.7.22 VoltageLevel

A collection of equipment at one common system voltage forming a switchgear. The equipment typically consist of breakers, busbars, instrumentation, control, regulation and protection devices as well as assemblies of all these.

DCConductingEquipment and DCConverterUnit is not allowed in VoltageLevels that is intended for AC equipment only. DCConductingEquipment is allowed in DCEquipmentContainers only. The ACDCConverter is ConductingEquipment. Other AC

equipment in a converter unit are the power transformer and converter reactor. All three shall be located in a DCConverterUnit and not a VoltageLevel.

Inheritance path: ->EquipmentContainer->ConnectivityNodeContainer->PowerSystemResource->IdentifiedObject

Table 176 shows all attributes of VoltageLevel.

Table 176 – Attributes of Core::VoltageLevel

Attribute name	mult.	Attribute type	Description
highVoltageLimit	0..1	Voltage	The bus bar's high voltage limit
lowVoltageLimit	0..1	Voltage	The bus bar's low voltage limit
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 177 shows all association ends of VoltageLevel with other classes.

Table 177 – Association ends of Core::VoltageLevel with other classes

mult.	mult.	Class type name	Description
Substation	[1..1]	Substation	The substation of the voltage level. Association Based On: Core:: Substation.Substation[1] ----- Core:: VoltageLevel.VoltageLevels[0..*]
BaseVoltage	[1..1]	BaseVoltage	The base voltage used for all equipment within the voltage level. Association Based On: Core:: VoltageLevel.VoltageLevel[0..*] ----- Core:: BaseVoltage.BaseVoltage[1]

6.7.23 Enumerations

6.7.23.1 CurveStyle

Style or shape of curve.

Literal	Description	Code
constantYValue	The Y-axis values are assumed constant until the next curve point and prior to the first curve point.	
straightLineYValues	The Y-axis values are assumed to be a straight line between values. Also known as linear interpolation.	

6.7.23.2 PhaseCode

Enumeration of phase identifiers. Allows designation of phases for both transmission and distribution equipment, circuits and loads. Residential and small commercial loads are often served from single-phase, or split-phase, secondary circuits. For example of s12N, phases 1

and 2 refer to hot wires that are 180 degrees out of phase, while N refers to the neutral wire. Through single-phase transformer connections, these secondary circuits may be served from one or two of the primary phases A, B, and C. For three-phase loads, use the A, B, C phase codes instead of s12N.

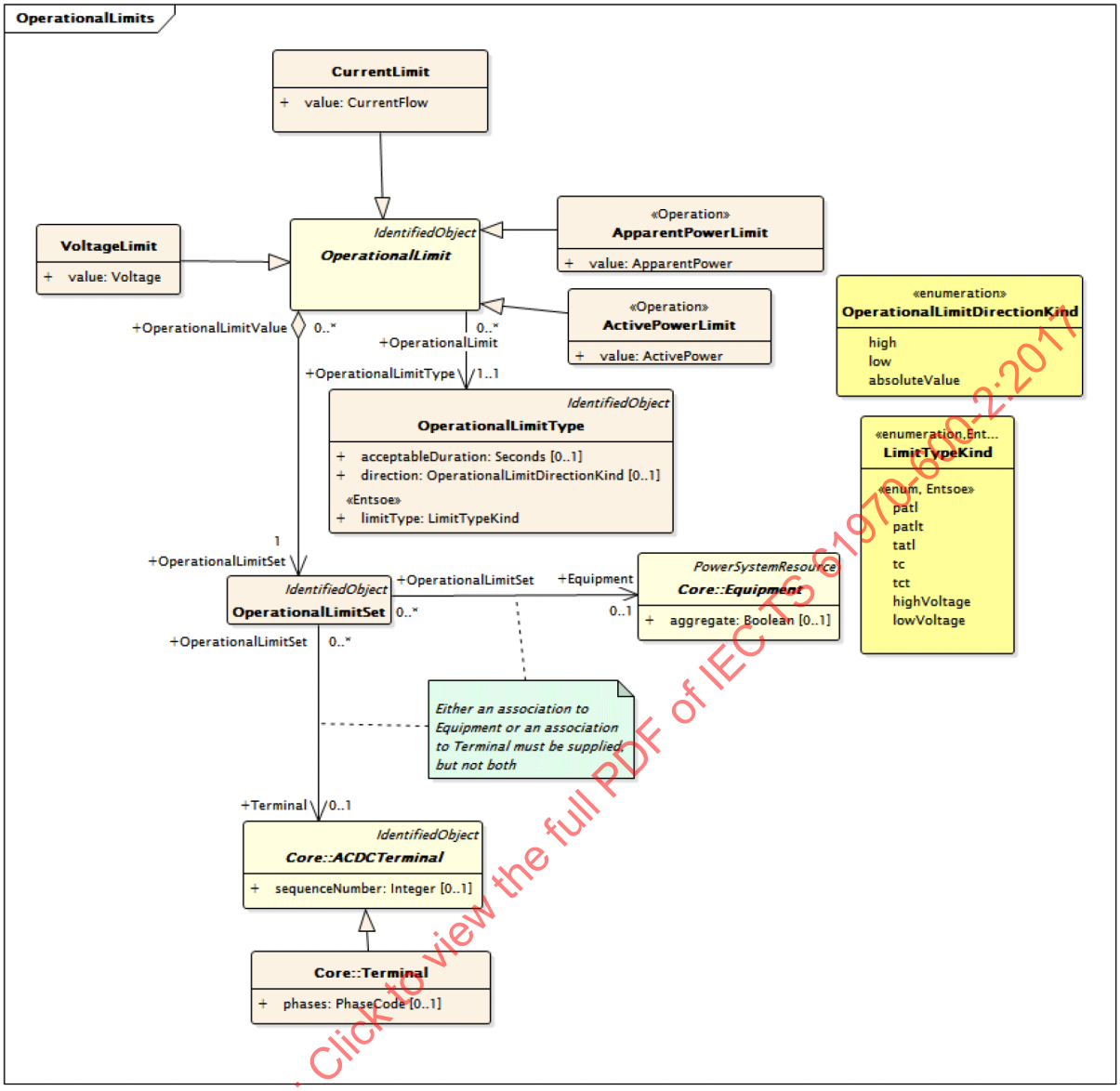
Literal	Description	Code
ABCN	Phases A, B, C, and N.	
ABC	Phases A, B, and C.	
ABN	Phases A, B, and neutral.	
ACN	Phases A, C and neutral.	
BCN	Phases B, C, and neutral.	
AB	Phases A and B.	
AC	Phases A and C.	
BC	Phases B and C.	
AN	Phases A and neutral.	
BN	Phases B and neutral.	
CN	Phases C and neutral.	
A	Phase A.	
B	Phase B.	
C	Phase C.	
N	Neutral phase.	
s1N	Secondary phase 1 and neutral.	
s2N	Secondary phase 2 and neutral.	
s12N	Secondary phases 1, 2, and neutral.	
s1	Secondary phase 1.	
s2	Secondary phase 2.	
s12	Secondary phase 1 and 2.	

6.8 OperationalLimits

6.8.1 General

The OperationalLimits package models a specification of limits associated with equipment and other operational entities.

Figure 16 shows the diagram.



IEC

Figure 16 – diagram OperationalLimits

Figure 17 shows the diagram.

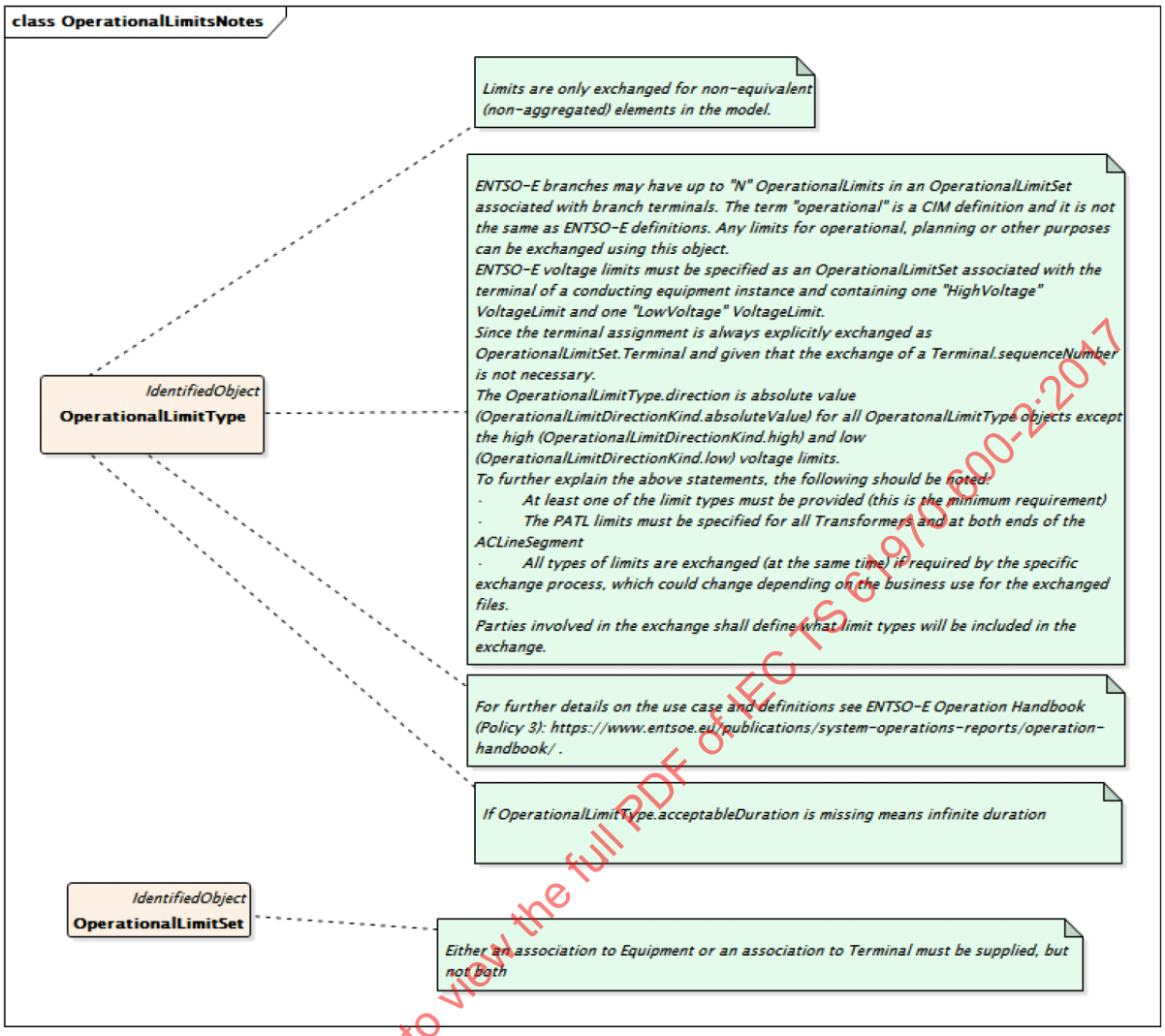


Figure 17 – diagram OperationalLimitsNotes

6.8.2 (Operation) ActivePowerLimit

Limit on active power flow.

Inheritance path: ->OperationalLimit->IdentifiedObject

Table 178 shows all attributes of ActivePowerLimit.

Table 178 – Attributes of OperationalLimits::ActivePowerLimit

Attribute name	mult.	Attribute type	Description
value (Operation)	1..1	ActivePower	Value of active power limit.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 179 shows all association ends of OperationalLimit with other classes.

Table 179 – Association ends of OperationalLimits::OperationalLimit with other classes

mult.	mult.	Class type name	Description
OperationalLimitSet	1..1	OperationalLimitSet	Inherited Association Based On: OperationalLimits:: OperationalLimit.OperationalLimitValue[0..*] --- OperationalLimits:: OperationalLimitSet.OperationalLimitSet[1]
OperationalLimitType	1..1	OperationalLimitType	Inherited Association Based On: OperationalLimits:: OperationalLimit.OperationalLimit[0..*] --- OperationalLimits:: OperationalLimitType.OperationalLimitType[0..1]

6.8.3 (Operation) ApparentPowerLimit

Apparent power limit.

Inheritance path: ->OperationalLimit->IdentifiedObject

Table 180 shows all attributes of ApparentPowerLimit.

Table 180 – Attributes of OperationalLimits::ApparentPowerLimit

Attribute name	mult.	Attribute type	Description
value (Operation)	1..1	ApparentPower	The apparent power limit.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 181 shows all association ends of OperationalLimit with other classes.

Table 181 – Association ends of OperationalLimits::OperationalLimit with other classes

mult.	mult.	Class type name	Description
OperationalLimitSet	1..1	OperationalLimitSet	Inherited Association Based On: OperationalLimits:: OperationalLimit.OperationalLimitValue[0..*] --- OperationalLimits:: OperationalLimitSet.OperationalLimitSet[1]
OperationalLimitType	1..1	OperationalLimitType	Inherited Association Based On: OperationalLimits:: OperationalLimit.OperationalLimit[0..*] --- OperationalLimits:: OperationalLimitType.OperationalLimitType[0..1]

6.8.4 CurrentLimit

Operational limit on current.

Inheritance path: ->OperationalLimit->IdentifiedObject

Table 182 shows all attributes of CurrentLimit.

Table 182 – Attributes of OperationalLimits::CurrentLimit

Attribute name	mult.	Attribute type	Description
value	1..1	CurrentFlow	Limit on current flow.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 183 shows all association ends of OperationalLimit with other classes.

Table 183 – Association ends of OperationalLimits::OperationalLimit with other classes

mult.	mult.	Class type name	Description
OperationalLimitSet	1..1	OperationalLimitSet	Inherited Association Based On: OperationalLimits:: OperationalLimit.OperationalLimitValue[0..*] --- OperationalLimits:: OperationalLimitSet.OperationalLimitSet[1]
OperationalLimitType	1..1	OperationalLimitType	Inherited Association Based On: OperationalLimits:: OperationalLimit.OperationalLimit[0..*] --- OperationalLimits:: OperationalLimitType.OperationalLimitType[0..1]

6.8.5 OperationalLimit(Abstract)

A value associated with a specific kind of limit.

The sub class value attribute shall be positive.

The sub class value attribute is inversely proportional to OperationalLimitType.acceptableDuration (acceptableDuration for short). A pair of value_x and acceptableDuration_x are related to each other as follows: if value_1 > value_2 > value_3 >... then acceptableDuration_1 < acceptableDuration_2 < acceptableDuration_3 < ...

A value_x with direction="high" shall be greater than a value_y with direction="low".

Inheritance path: ->IdentifiedObject

Table 184 shows all attributes of OperationalLimit.

Table 184 – Attributes of OperationalLimits::OperationalLimit

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 185 shows all association ends of OperationalLimit with other classes.

Table 185 – Association ends of OperationalLimits::OperationalLimit with other classes

mult.	mult.	Class type name	Description
OperationalLimitSet	[1..1]	OperationalLimitSet	Values of equipment limits. Association Based On: OperationalLimits:: OperationalLimit.OperationalLimitValue[0..*] ----- OperationalLimits:: OperationalLimitSet.OperationalLimitSet[1]
OperationalLimitType	[1..1]	OperationalLimitType	The limit type associated with this limit. Association Based On: OperationalLimits:: OperationalLimit.OperationalLimit[0..*] ----- OperationalLimits:: OperationalLimitType.OperationalLimitType[0..1]

6.8.6 OperationalLimitSet

A set of limits associated with equipment. Sets of limits might apply to a specific temperature, or season for example. A set of limits may contain different severities of limit levels that would apply to the same equipment. The set may contain limits of different types such as apparent power and current limits or high and low voltage limits that are logically applied together as a set.

Either an association to Equipment or an association to Terminal must be supplied, but not both.

Inheritance path: ->IdentifiedObject

Table 186 shows all attributes of OperationalLimitSet.

Table 186 – Attributes of OperationalLimits::OperationalLimitSet

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 187 shows all association ends of OperationalLimitSet with other classes.

Table 187 – Association ends of OperationalLimits::OperationalLimitSet with other classes

mult.	mult.	Class type name	Description
Equipment	[0..1]	Equipment	The equipment to which the limit set applies. Association Based On: OperationalLimits:: OperationalLimitSet.OperationalLimitSet[0..*] ----- Core:: Equipment.Equipment[0..1]
Terminal	[0..1]	ACDCTerminal	Association Based On: OperationalLimits:: OperationalLimitSet.OperationalLimitSet[0..*] ----- Core:: ACDCTerminal.Terminal[0..1]

6.8.7 OperationalLimitType

The operational meaning of a category of limits.

If OperationalLimitType.acceptableDuration is missing means infinite duration

Limits are only exchanged for non-equivalent (non-aggregated) elements in the model.

For further details on the use case and definitions see ENTSO-E Operation Handbook (Policy 3): https://www.entsoe.eu/publications/system-operations-reports/operation-handbook/ .

ENTSO-E branches may have up to "N" OperationalLimits in an OperationalLimitSet associated with branch terminals. The term "operational" is a CIM definition and it is not the same as ENTSO-E definitions. Any limits for operational, planning or other purposes can be exchanged using this object. ENTSO-E voltage limits must be specified as an OperationalLimitSet associated with the terminal of a conducting equipment instance and containing one "HighVoltage" VoltageLimit and one "LowVoltage" VoltageLimit. Since the terminal assignment is always explicitly exchanged as OperationalLimitSet.Terminal and given that the exchange of a Terminal.sequenceNumber is not necessary. The OperationalLimitType.direction is absolute value (OperationalLimitDirectionKind.absoluteValue) for all OperatonalLimitType objects except the high (OperationalLimitDirectionKind.high) and low (OperationalLimitDirectionKind.low) voltage limits. To further explain the above statements, the following should be noted: At least one of the limit types must be provided (this is the minimum requirement). The PATL limits must be specified for all Transformers and at both ends of the ACLineSegment. All types of limits are exchanged (at the same time) if required by the specific exchange process, which could change depending on the business use for the exchanged files. Parties involved in the exchange shall define what limit types will be included in the exchange.

Inheritance path: ->IdentifiedObject

Table 188 shows all attributes of OperationalLimitType.

Table 188 – Attributes of OperationalLimits::OperationalLimitType

Attribute name	mult.	Attribute type	Description
acceptableDuration	0..1	Seconds	The nominal acceptable duration of the limit. Limits are commonly expressed in terms of the a time limit for which the limit is normally acceptable. The actual acceptable duration of a specific limit may depend on other local factors such as temperature or wind speed.
limitType (Entsoe)	1..1	LimitTypeKind	Types of limits defined in the ENTSO-E Operational Handbook Policy 3.
direction	0..1	OperationalLimitDirectionKind	The direction of the limit.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.8.8 VoltageLimit

Operational limit applied to voltage.

Inheritance path: ->OperationalLimit->IdentifiedObject

Table 189 shows all attributes of VoltageLimit.

Table 189 – Attributes of OperationalLimits::VoltageLimit

Attribute name	mult.	Attribute type	Description
value	1..1	Voltage	Limit on voltage. High or low limit nature of the limit depends upon the properties of the operational limit type.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 190 shows all association ends of OperationalLimit with other classes.

Table 190 – Association ends of OperationalLimits::OperationalLimit with other classes

mult.	mult.	Class type name	Description
OperationalLimitSet	1..1	OperationalLimitSet	Inherited Association Based On: OperationalLimits:: OperationalLimit.OperationalLimitValue[0..*] ----- OperationalLimits:: OperationalLimitSet.OperationalLimitSet[1]
OperationalLimitType	1..1	OperationalLimitType	Inherited Association Based On: OperationalLimits:: OperationalLimit.OperationalLimit[0..*] ----- OperationalLimits:: OperationalLimitType.OperationalLimitType[0..1]

6.8.9 Enumerations

6.8.9.1 LimitTypeKind

The enumeration defines the kinds of the limit types.

Literal	Description	Code
patl	The Permanent Admissible Transmission Loading (PATL) is the loading in Amps, MVA or MW that can be accepted by a network branch for an unlimited duration without any risk for the material. The duration attribute is not used and shall be excluded for the PATL limit type. Hence only one limit value exists for the PATL type.	
patlt	Permanent Admissible Transmission Loading Threshold (PATLT) is a value in engineering units defined for PATL and calculated using percentage less than 100 of the PATL type intended to alert operators of an arising condition. The percentage should be given in the name of the OperationalLimitSet. The aceptableDuration is another way to express the severity of the limit.	
tatl	Temporarily Admissible Transmission Loading (TATL) which is the loading in Amps, MVA or MW that can be accepted by a branch for a certain limited duration. The TATL can be defined in different ways: as a fixed percentage of the PATL for a given time (for example, 115% of the PATL that can be accepted during 15 minutes), pairs of TATL type and Duration calculated for each line taking into account its particular configuration and conditions of functioning (for example, it can define a TATL acceptable during 20 minutes and another one acceptable during 10 minutes). Such a definition of TATL can depend on the initial operating conditions of the network element (sag situation of a line). The duration attribute can be used define several TATL limit types. Hence multiple TATL limit values may exist having different durations.	
tc	Tripping Current (TC) is the ultimate intensity without any delay. It is defined as the threshold the line will trip without any possible remedial actions. The tripping of the network element is ordered by protections against short circuits or by overload protections, but in any case, the activation delay of these protections is not compatible with the reaction delay of an operator (less than one minute). The duration is always zero and the duration attribute may be left out. Hence only one limit value exists for the TC type.	
tct	Tripping Current Threshold (TCT) is a value in engineering units defined for TC and calculated using percentage less than 100 of the TC type intended to alert operators of an arising condition. The percentage should be given in the name of the OperationalLimitSet. The aceptableDuration is another way to express the severity of the limit.	
highVoltage	Referring to the rating of the equipments, a voltage too high can lead to accelerated ageing or the destruction of the equipment. This limit type may or may not have duration.	
lowVoltage	A too low voltage can disturb the normal operation of some protections and transformer equipped with on-load tap changers, electronic power devices or can affect the behaviour of the auxiliaries of generation units. This limit type may or may not have duration.	

6.8.9.2 OperationalLimitDirectionKind

The direction attribute describes the side of a limit that is a violation.

Literal	Description	Code
high	High means that a monitored value above the limit value is a violation. If applied to a terminal flow, the positive direction is into the terminal.	
low	Low means a monitored value below the limit is a violation. If applied to a terminal flow, the positive direction is into the terminal.	
absoluteValue	An absoluteValue limit means that a monitored absolute value above the limit value is a violation.	

6.9 Wires

6.9.1 General

An extension to the Core and Topology package that models information on the electrical characteristics of Transmission and Distribution networks. This package is used by network applications such as State Estimation, Load Flow and Optimal Power Flow.

Figure 18 shows the diagram.

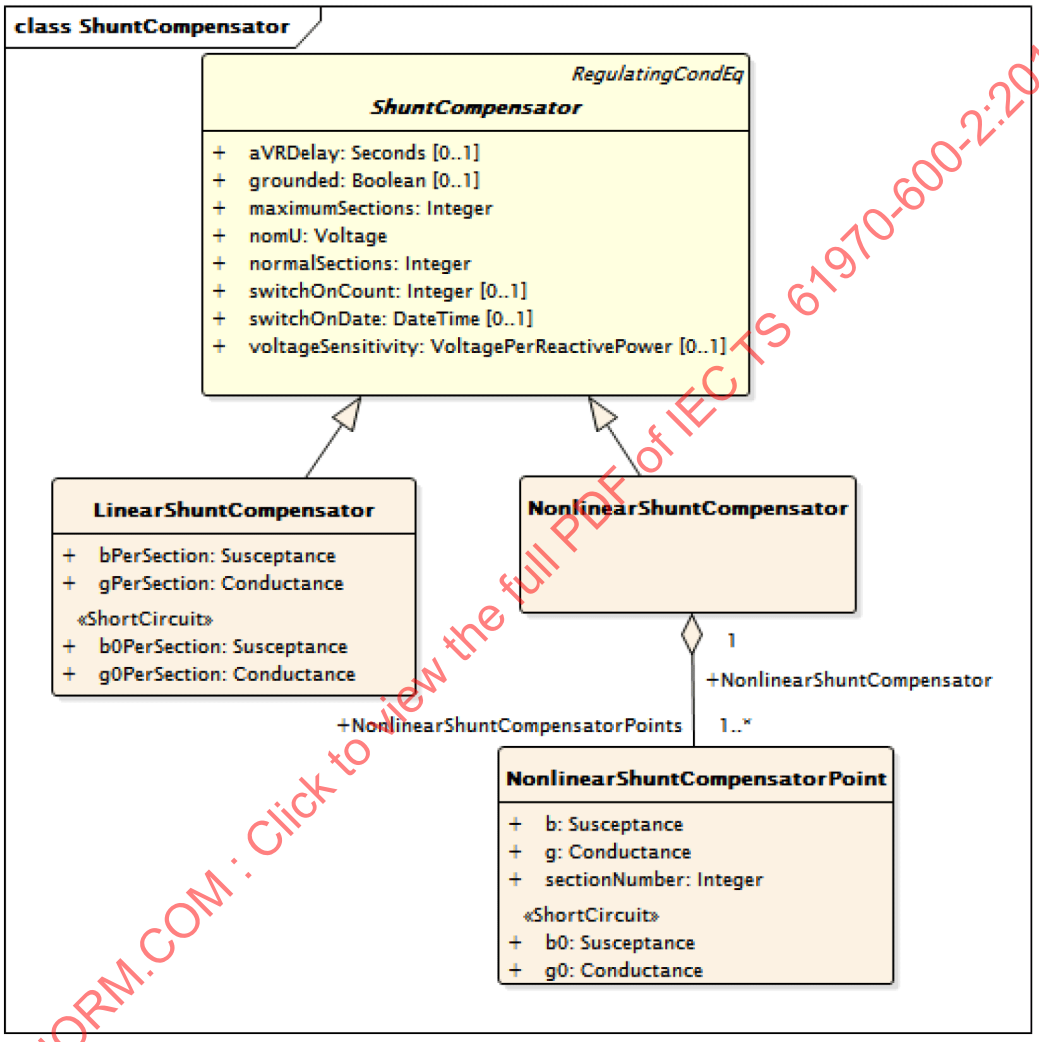
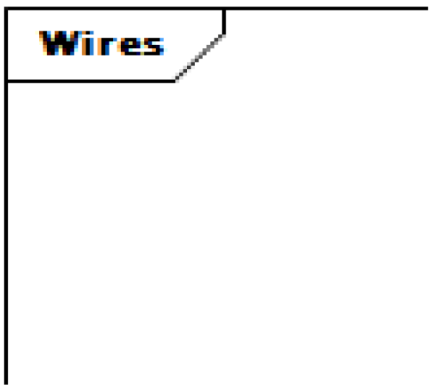


Figure 18 – diagram ShuntCompensator

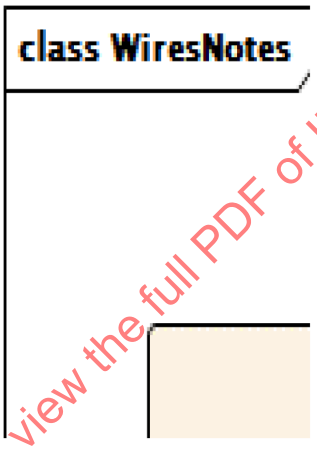
Figure 19 shows the diagram.



IEC

Figure 19 – diagram Wires

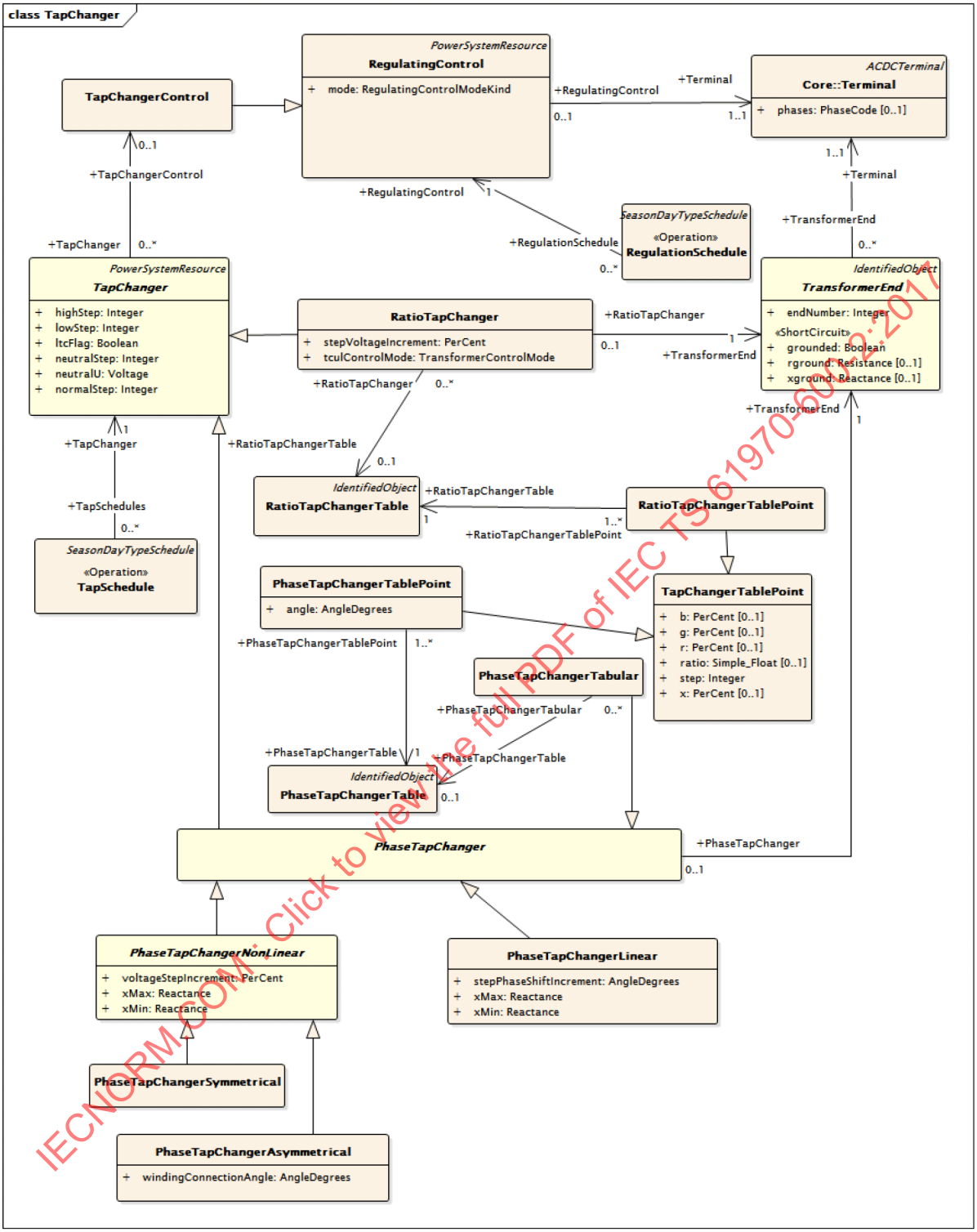
Figure 20 shows the diagram.



IEC

Figure 20 – diagram WiresNotes

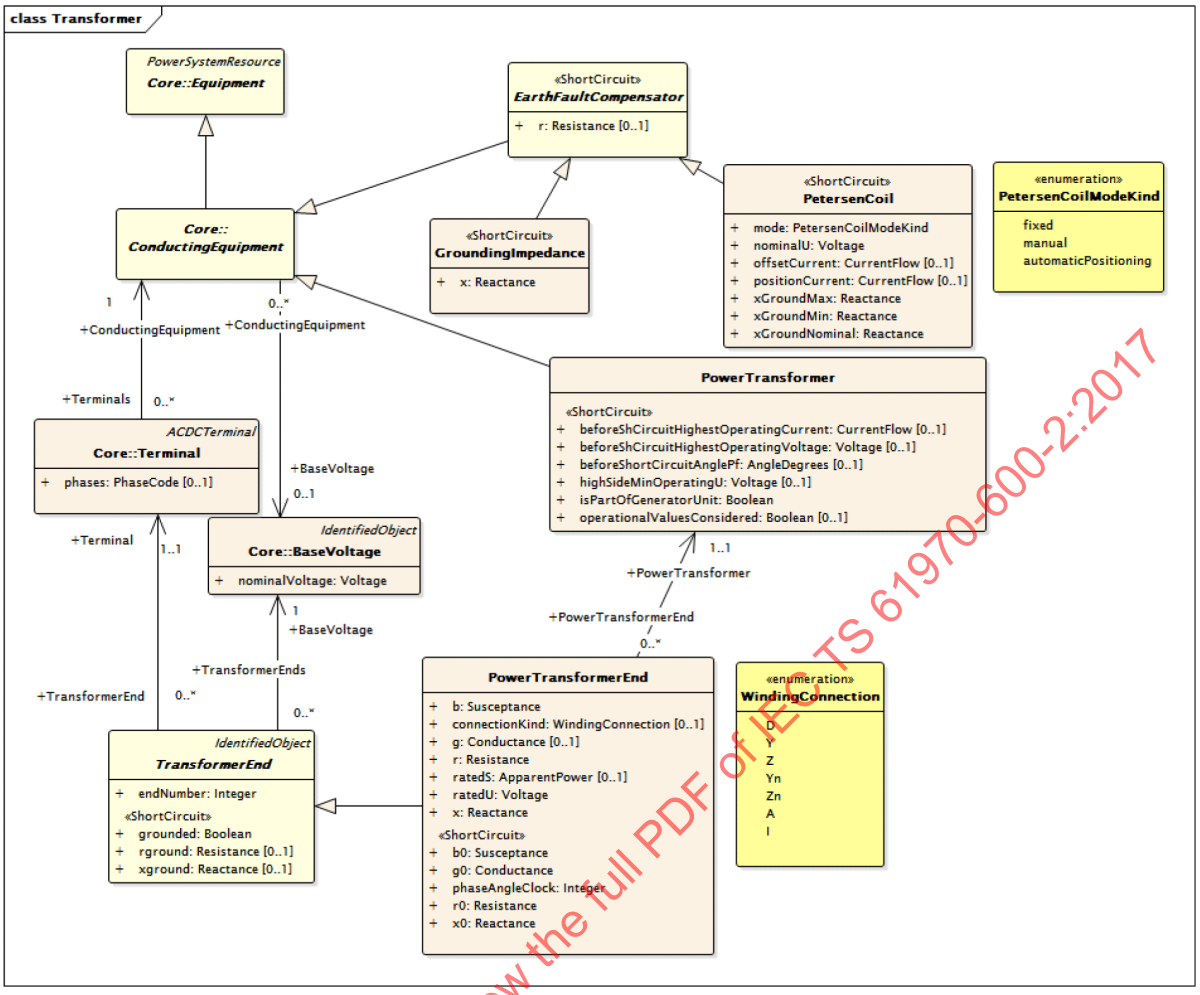
Figure 21 shows the diagram.



IEC

Figure 21 – diagram TapChanger

Figure 22 shows the diagram.



IEC

Figure 22 – diagram Transformer

6.9.2 ACLineSegment

A wire or combination of wires, with consistent electrical characteristics, building a single electrical system, used to carry alternating current between points in the power system. For symmetrical, transposed 3ph lines, it is sufficient to use attributes of the line segment, which describe impedances and admittances for the entire length of the segment. Additionally impedances can be computed by using length and associated per length impedances. The BaseVoltage at the two ends of ACLineSegments in a Line shall have the same BaseVoltage.nominalVoltage. However, boundary lines may have slightly different BaseVoltage.nominalVoltages and variation is allowed. Larger voltage difference in general requires use of an equivalent branch.

OCL constraint: An ACLineSegment must have a BaseVoltage.

Each ACLineSegment is required to have an association to a BaseVoltage. The association to Line is not required.

The positive sequence resistance (r) and reactance (x) are the same as negative sequence. Therefore negative sequence data is not modeled.

ENTSO-E exchanges allow 10 % difference of the BaseVoltage.nominalVoltage at the two ends of an ACLineSegment representing a complete tie-line or connecting to a boundary node.

Using the “EquipmentContainer” association, an ACLineSegment can only be contained by a Line, but the association to Line is not required.

Inheritance path: ->Conductor->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 191 shows all attributes of ACLineSegment.

Table 191 – Attributes of Wires::ACLineSegment

Attribute name	mult.	Attribute type	Description
b0ch (ShortCircuit)	1..1	Susceptance	Zero sequence shunt (charging) susceptance, uniformly distributed, of the entire line section.
bch	1..1	Susceptance	Positive sequence shunt (charging) susceptance, uniformly distributed, of the entire line section. This value represents the full charging over the full length of the line.
g0ch (ShortCircuit)	1..1	Conductance	Zero sequence shunt (charging) conductance, uniformly distributed, of the entire line section.
gch	0..1	Conductance	Positive sequence shunt (charging) conductance, uniformly distributed, of the entire line section.
r	1..1	Resistance	Positive sequence series resistance of the entire line section.
r0 (ShortCircuit)	1..1	Resistance	Zero sequence series resistance of the entire line section.
shortCircuitEndTemperature (ShortCircuit)	1..1	Temperature	Maximum permitted temperature at the end of SC for the calculation of minimum short-circuit currents. Used for short circuit data exchange according to IEC 60909
x	1..1	Reactance	Positive sequence series reactance of the entire line section.
x0 (ShortCircuit)	1..1	Reactance	Zero sequence series reactance of the entire line section.
length	0..1	Length	see Conductor
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 192 shows all association ends of ConductingEquipment with other classes.

Table 192 – Association ends of Core::ConductingEquipment with other classes

mult.	mult.	Class type name	Description
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.3 AsynchronousMachine

A rotating machine whose shaft rotates asynchronously with the electrical field. Also known as an induction machine with no external connection to the rotor windings, e.g squirrel-cage induction machine.

The attribute rxLockedRotorRatio is an optional attribute even if short circuit data is exchanged because IEC 60909 defines default values depending on motor size.

Inheritance path: ->RotatingMachine->RegulatingCondEq->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 193 shows all attributes of AsynchronousMachine.

Table 193 – Attributes of Wires::AsynchronousMachine

Attribute name	mult.	Attribute type	Description
converterFedDrive (ShortCircuit)	1..1	Boolean	Indicates whether the machine is a converter fed drive. Used for short circuit data exchange according to IEC 60909
efficiency (ShortCircuit)	1..1	PerCent	Efficiency of the asynchronous machine at nominal operation in percent. Indicator for converter drive motors. Used for short circuit data exchange according to IEC 60909
iaIrRatio (ShortCircuit)	1..1	Simple_Float	Ratio of locked-rotor current to the rated current of the motor (Ia/Ir). Used for short circuit data exchange according to IEC 60909
nominalFrequency	0..1	Frequency	Nameplate data indicates if the machine is 50 or 60 Hz.
nominalSpeed	0..1	RotationSpeed	Nameplate data. Depends on the slip and number of pole pairs.
polePairNumber (ShortCircuit)	1..1	Integer	Number of pole pairs of stator. Used for short circuit data exchange according to IEC 60909
ratedMechanicalPower (ShortCircuit)	1..1	ActivePower	Rated mechanical power (Pr in the IEC 60909-0). Used for short circuit data exchange according to IEC 60909.
reversible (ShortCircuit)	1..1	Boolean	Indicates for converter drive motors if the power can be reversible. Used for short circuit data exchange according to IEC 60909
rxLockedRotorRatio (ShortCircuit)	0..1	Simple_Float	Locked rotor ratio (R/X). Used for short circuit data exchange according to IEC 60909
ratedPowerFactor	0..1	Simple_Float	see RotatingMachine
ratedS	0..1	ApparentPower	see RotatingMachine
ratedU	0..1	Voltage	see RotatingMachine
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 194 shows all association ends of RotatingMachine with other classes.

Table 194 – Association ends of Wires::RotatingMachine with other classes

mult.	mult.	Class type name	Description
GeneratingUnit	0..1	GeneratingUnit	Inherited Association Based On: Production:: GeneratingUnit.GeneratingUnit[0..1] --- Wires:: RotatingMachine.RotatingMachine[1..*]
RegulatingControl	0..1	RegulatingControl	Inherited Association Based On: Wires:: RegulatingCondEq.RegulatingCondEq[0..*] --- Wires:: RegulatingControl.RegulatingControl[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] --- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] --- Core:: Equipment.Equipments[0..*]

6.9.4 Breaker

A mechanical switching device capable of making, carrying, and breaking currents under normal circuit conditions and also making, carrying for a specified time, and breaking currents under specified abnormal circuit conditions e.g. those of short circuit.

For switching Devices, Naming Convention and main containership hierarchy is: GeographicalRegion/SubGeographicalRegion/Substation/VoltageLevel or GeographicalRegion/SubGeographicalRegion/SubstationàVoltageLevel/Bay

Inheritance path: ->ProtectedSwitch->Switch->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 195 shows all attributes of Breaker.

Table 195 – Attributes of Wires::Breaker

Attribute name	mult.	Attribute type	Description
normalOpen	1..1	Boolean	see Switch
ratedCurrent	0..1	CurrentFlow	see Switch
retained	1..1	Boolean	see Switch
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 196 shows all association ends of ConductingEquipment with other classes.

Table 196 – Association ends of Core::ConductingEquipment with other classes

mult.	mult.	Class type name	Description
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.5 BusbarSection

A conductor, or group of conductors, with negligible impedance, that serve to connect other conducting equipment within a single substation. Voltage measurements are typically obtained from VoltageTransformers that are connected to busbar sections. A bus bar section may have many physical terminals but for analysis is modelled with exactly one logical terminal.

BusBarSection could be associated to only 1 Terminal.

The attribute ipMax is an optional attribute even if short circuit data is exchanged, as not always entered by the user (e.g. the example test model described in 6.2 of IEC TR 60909-4:2000 does not include these values.

Naming Convention and main containership hierarchy is: GeographicalRegion/SubGeographicalRegion/Substation/VoltageLevel

Inheritance path: ->Connector->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 197 shows all attributes of BusbarSection.

Table 197 – Attributes of Wires::BusbarSection

Attribute name	mult.	Attribute type	Description
ipMax (ShortCircuit)	0..1	CurrentFlow	Maximum allowable peak short-circuit current of busbar (Ipmax in the IEC 60909-0). Mechanical limit of the busbar in the substation itself. Used for short circuit data exchange according to IEC 60909
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 198 shows all association ends of ConductingEquipment with other classes.

Table 198 – Association ends of Core::ConductingEquipment with other classes

mult.	mult.	Class type name	Description
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.6 Conductor(Abstract)

Combination of conducting material with consistent electrical characteristics, building a single electrical system, used to carry current between points in the power system.

Inheritance path: ->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 199 shows all attributes of Conductor.

Table 199 – Attributes of Wires::Conductor

Attribute name	mult.	Attribute type	Description
length	0..1	Length	Segment length for calculating line section capabilities
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 200 shows all association ends of ConductingEquipment with other classes.

Table 200 – Association ends of Core::ConductingEquipment with other classes

mult.	mult.	Class type name	Description
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.7 Connector(Abstract)

A conductor, or group of conductors, with negligible impedance, that serve to connect other conducting equipment within a single substation and are modelled with a single logical terminal.

Inheritance path: ->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 201 shows all attributes of Connector.

Table 201 – Attributes of Wires::Connector

Attribute name	mult.	Attribute type	Description
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 202 shows all association ends of ConductingEquipment with other classes.

Table 202 – Association ends of Core::ConductingEquipment with other classes

mult.	mult.	Class type name	Description
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.8 Disconnector

A manually operated or motor operated mechanical switching device used for changing the connections in a circuit, or for isolating a circuit or equipment from a source of power. It is required to open or close circuits when negligible current is broken or made.

For switching Devices, Naming Convention and main containership hierarchy is: GeographicalRegion/SubGeographicalRegion/Substation/VoltageLevel or GeographicalRegion/SubGeographicalRegion/SubstationàVoltageLevel/Bay

Inheritance path: ->Switch->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 203 shows all attributes of Disconnector.

Table 203 – Attributes of Wires::Disconnector

Attribute name	mult.	Attribute type	Description
normalOpen	1..1	Boolean	see Switch
ratedCurrent	0..1	CurrentFlow	see Switch
retained	1..1	Boolean	see Switch
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 204 shows all association ends of ConductingEquipment with other classes.

Table 204 – Association ends of Core::ConductingEquipment with other classes

mult.	mult.	Class type name	Description
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ---- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.9 (ShortCircuit) EarthFaultCompensator(Abstract)

A conducting equipment used to represent a connection to ground which is typically used to compensate earth faults. An earth fault compensator device modeled with a single terminal implies a second terminal solidly connected to ground. If two terminals are modeled, the ground is not assumed and normal connection rules apply.

The GroundingImpedance and PetersenCoil connect the same way as other devices using Terminal. The phases attribute on Terminal indicates the connection is at the neutral phase. If a PowerTransformer has a Y connection on one terminal (secondary) and in case it is necessary to connect a PetersenCoil, the PowerTransformer's secondary Terminal where the coil is connected should be specified with phases attribute equal to ABCN and the PetersenCoil's Terminal should be specified with phase N. This allows the ACLineSegment to connect with phases ABC (the default) and the PetersenCoil to connect to the same TransformerTerminal but only on the N phase. All EarthFaultCompensators (GroundingImpedance and PetersenCoil) are expected to have only one Terminal and be solidly connected to ground on the other unmodeled terminal. All PetersenCoil terminals are expected to have only phase N. Basically a normal modeling of non-neutral network is done and then added to the Petersen coil model which includes: 1) Adding the phasing of ABCN to the transformer secondary terminal so there is a place to connect neutral 2) Adding the PetersenCoil object with its associated Terminal which is phase N, since only one Terminal is modeled the PetersenCoil is assumed solidly grounded 3) Associating the PetersenCoil's single Terminal to the ConnectivityNode at transformer secondary.

Inheritance path: ->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 205 shows all attributes of EarthFaultCompensator.

Table 205 – Attributes of Wires::EarthFaultCompensator

Attribute name	mult.	Attribute type	Description
r (ShortCircuit)	0..1	Resistance	Nominal resistance of device.
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 206 shows all association ends of ConductingEquipment with other classes.

Table 206 – Association ends of Core::ConductingEquipment with other classes

mult.	mult.	Class type name	Description
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.10 EnergyConsumer

Generic user of energy – a point of consumption on the power system model.

Attributes (pfixed, qfixed, pfixedPct and qfixedPct) are used for load allocation. Attributes (pfixed and qfixed) represent base load, while attributes (pfixedPct and qfixedPct) represent the time-varying components.

- The definition of the real and reactive power injections for an EnergyConsumer can be done using different sets of attributes. In the simplest case, the injections can be defined directly using only the attributes “pfixed” and “qfixed”.
- To specify conforming and nonconforming loads, the classes ConformLoad, NonConformLoad, or their subtypes should be used.
- The attributes defining the affect of voltage and frequency on the injection defined by an associated LoadResponseCharacteristic should be supplied, if they are available, but are not required.

Naming Convention and main containership hierarchy is: GeographicalRegion/SubGeographicalRegion/Substation/VoltageLevel

If LoadResponseCharacteristic is missing, this load is assumed to be constant power.

Inheritance path: ->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 207 shows all attributes of EnergyConsumer.

Table 207 – Attributes of Wires::EnergyConsumer

Attribute name	mult.	Attribute type	Description
pfixed (Operation)	0..1	ActivePower	Active power of the load that is a fixed quantity. Load sign convention is used, i.e. positive sign means flow out from a node.
pfixedPct (Operation)	0..1	PerCent	Fixed active power as per cent of load group fixed active power. Load sign convention is used, i.e. positive sign means flow out from a node.
qfixed (Operation)	0..1	ReactivePower	Reactive power of the load that is a fixed quantity. Load sign convention is used, i.e. positive sign means flow out from a node.
qfixedPct (Operation)	0..1	PerCent	Fixed reactive power as per cent of load group fixed reactive power. Load sign convention is used, i.e. positive sign means flow out from a node.
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 208 shows all association ends of EnergyConsumer with other classes.

Table 208 – Association ends of Wires::EnergyConsumer with other classes

mult.	mult.	Class type name	Description
LoadResponse	[0..1]	LoadResponseCharacteristic	The load response characteristic of this load. If missing, this load is assumed to be constant power. Association Based On: Wires:: EnergyConsumer.EnergyConsumer[0..*] ----- LoadModel:: LoadResponseCharacteristic.LoadResponse[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.11 ExternalNetworkInjection

This class represents external network and it is used for IEC 60909 calculations.

The attributes ikSecond and voltageFactor are optional attributes even if short circuit data is exchanged. These attributes are used only if short circuit calculations are done according to superposition method.

Inheritance path: ->RegulatingCondEq->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 209 shows all attributes of ExternalNetworkInjection.

Table 209 – Attributes of Wires::ExternalNetworkInjection

Attribute name	mult.	Attribute type	Description
governorSCD	1..1	ActivePowerPerFrequency	Power Frequency Bias. This is the change in power injection divided by the change in frequency and negated. A positive value of the power frequency bias provides additional power injection upon a drop in frequency.
ikSecond (ShortCircuit)	0..1	Boolean	Indicates whether initial symmetrical short-circuit current and power have been calculated according to IEC (Ik").
maxInitialSymShCCurrent (ShortCircuit)	1..1	CurrentFlow	Maximum initial symmetrical short-circuit currents (Ik" max) in A (Ik" = Sk"/(SQRT(3) Un)). Used for short circuit data exchange according to IEC 60909
maxP	1..1	ActivePower	Maximum active power of the injection.
maxQ	1..1	ReactivePower	Not for short circuit modelling; It is used for modelling of infeed for load flow exchange. If maxQ and minQ are not used ReactiveCapabilityCurve can be used
maxR0ToX0Ratio (ShortCircuit)	1..1	Simple_Float	Maximum ratio of zero sequence resistance of Network Feeder to its zero sequence reactance (R(0)/X(0) max). Used for short circuit data exchange according to IEC 60909
maxR1ToX1Ratio (ShortCircuit)	1..1	Simple_Float	Maximum ratio of positive sequence resistance of Network Feeder to its positive sequence reactance (R(1)/X(1) max). Used for short circuit data exchange according to IEC 60909
maxZ0ToZ1Ratio (ShortCircuit)	1..1	Simple_Float	Maximum ratio of zero sequence impedance to its positive sequence impedance (Z(0)/Z(1) max). Used for short circuit data exchange according to IEC 60909
minInitialSymShCCurrent (ShortCircuit)	1..1	CurrentFlow	Minimum initial symmetrical short-circuit currents (Ik" min) in A (Ik" = Sk"/(SQRT(3) Un)). Used for short circuit data exchange according to IEC 60909
minP	1..1	ActivePower	Minimum active power of the injection.
minQ	1..1	ReactivePower	Not for short circuit modelling; It is used for modelling of infeed for load flow exchange. If maxQ and minQ are not used ReactiveCapabilityCurve can be used
minR0ToX0Ratio (ShortCircuit)	1..1	Simple_Float	Indicates whether initial symmetrical short-circuit current and power have been calculated according to IEC (Ik"). Used for short circuit data exchange according to IEC 6090
minR1ToX1Ratio (ShortCircuit)	1..1	Simple_Float	Minimum ratio of positive sequence resistance of Network Feeder to its positive sequence reactance (R(1)/X(1) min). Used for short circuit data exchange according to IEC 60909

Attribute name	mult.	Attribute type	Description
minZ0ToZ1Ratio (ShortCircuit)	1..1	Simple_Float	Minimum ratio of zero sequence impedance to its positive sequence impedance (Z(0)/Z(1) min). Used for short circuit data exchange according to IEC 60909
voltageFactor (ShortCircuit)	0..1	PU	Voltage factor in pu, which was used to calculate short-circuit current Ik" and power Sk".
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 210 shows all association ends of RegulatingCondEq with other classes.

Table 210 – Association ends of Wires::RegulatingCondEq with other classes

mult.	mult.	Class type name	Description
RegulatingControl	0..1	RegulatingControl	Inherited Association Based On: Wires:: RegulatingCondEq.RegulatingCondEq[0..*] ----- Wires:: RegulatingControl.RegulatingControl[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.12 (ShortCircuit,Operation) Ground

A point where the system is grounded used for connecting conducting equipment to ground. The power system model can have any number of grounds.

Inheritance path: ->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 211 shows all attributes of Ground.

Table 211 – Attributes of Wires::Ground

Attribute name	mult.	Attribute type	Description
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 212 shows all association ends of ConductingEquipment with other classes.

Table 212 – Association ends of Core::ConductingEquipment with other classes

mult.	mult.	Class type name	Description
BaseVoltage	0..1	`BaseVoltage`	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	`EquipmentContainer`	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.13 (ShortCircuit,Operation) GroundDisconnector

A manually operated or motor operated mechanical switching device used for isolating a circuit or equipment from ground.

Inheritance path: ->Switch->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 213 shows all attributes of GroundDisconnector.

Table 213 – Attributes of Wires::GroundDisconnector

Attribute name	mult.	Attribute type	Description
normalOpen	1..1	Boolean	see Switch
ratedCurrent	0..1	CurrentFlow	see Switch
retained	1..1	Boolean	see Switch
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 214 shows all association ends of ConductingEquipment with other classes.

Table 214 – Association ends of Core::ConductingEquipment with other classes

mult.	mult.	Class type name	Description
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] --- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] --- Core:: Equipment.Equipments[0..*]

6.9.14 (ShortCircuit) GroundingImpedance

A fixed impedance device used for grounding.

Inheritance path: ->EarthFaultCompensator->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 215 shows all attributes of GroundingImpedance.

Table 215 – Attributes of Wires::GroundingImpedance

Attribute name	mult.	Attribute type	Description
x (ShortCircuit)	1..1	Reactance	Reactance of device.
r	0..1	Resistance	see EarthFaultCompensator
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 216 shows all association ends of ConductingEquipment with other classes.

Table 216 – Association ends of Core::ConductingEquipment with other classes

mult.	mult.	Class type name	Description
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] --- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.15 Junction

A point where one or more conducting equipments are connected with zero resistance.

Inheritance path: ->Connector->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 217 shows all attributes of Junction.

Table 217 – Attributes of Wires::Junction

Attribute name	mult.	Attribute type	Description
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 218 shows all association ends of ConductingEquipment with other classes.

Table 218 – Association ends of Core::ConductingEquipment with other classes

mult.	mult.	Class type name	Description
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.16 Line

Contains equipment beyond a substation belonging to a power transmission line.

Use of the Line class is not required. If used, it can only be used as a container for ACLineSegments and SeriesCompensators.

A Line is not required to be associated with a SubGeographicalRegion.

Inheritance path: ->EquipmentContainer->ConnectivityNodeContainer->PowerSystemResource->IdentifiedObject

Table 219 shows all attributes of Line.

Table 219 – Attributes of Wires::Line

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 220 shows all association ends of Line with other classes.

Table 220 – Association ends of Wires::Line with other classes

mult.	mult.	Class type name	Description
Region	[0..1]	SubGeographicalRegion	The sub-geographical region of the line. Association Based On: Core:: SubGeographicalRegion.Region[0..1] ----- Wires:: Line.Lines[0..*]

6.9.17 LinearShuntCompensator

A linear shunt compensator has banks or sections with equal admittance values.

Inheritance path: ->ShuntCompensator->RegulatingCondEq->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 221 shows all attributes of LinearShuntCompensator.

Table 221 – Attributes of Wires::LinearShuntCompensator

Attribute name	mult.	Attribute type	Description
b0PerSection (ShortCircuit)	1..1	Susceptance	Zero sequence shunt (charging) susceptance per section
bPerSection	1..1	Susceptance	Positive sequence shunt (charging) susceptance per section
g0PerSection (ShortCircuit)	1..1	Conductance	Zero sequence shunt (charging) conductance per section
gPerSection	1..1	Conductance	Positive sequence shunt (charging) conductance per section
aVRDelay	0..1	Seconds	see ShuntCompensator
grounded	0..1	Boolean	see ShuntCompensator
maximumSections	1..1	Integer	see ShuntCompensator
nomU	1..1	Voltage	see ShuntCompensator
normalSections	1..1	Integer	see ShuntCompensator
switchOnCount	0..1	Integer	see ShuntCompensator
switchOnDate	0..1	DateTime	see ShuntCompensator
voltageSensitivity	0..1	VoltagePerReactivePower	see ShuntCompensator
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 222 shows all association ends of RegulatingCondEq with other classes.

Table 222 – Association ends of Wires::RegulatingCondEq with other classes

mult.	mult.	Class type name	Description
RegulatingControl	0..1	RegulatingControl	Inherited Association Based On: Wires:: RegulatingCondEq.RegulatingCondEq[0..*] ----- Wires:: RegulatingControl.RegulatingControl[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.18 LoadBreakSwitch

A mechanical switching device capable of making, carrying, and breaking currents under normal operating conditions.

For switching Devices, Naming Convention and main containership hierarchy is:
GeographicalRegion/SubGeographicalRegion/Substation/VoltageLevel or
GeographicalRegion/SubGeographicalRegion/SubstationàVoltageLevel/Bay

Inheritance path: ->ProtectedSwitch->Switch->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 223 shows all attributes of LoadBreakSwitch.

Table 223 – Attributes of Wires::LoadBreakSwitch

Attribute name	mult.	Attribute type	Description
normalOpen	1..1	Boolean	see Switch
ratedCurrent	0..1	CurrentFlow	see Switch
retained	1..1	Boolean	see Switch
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 224 shows all association ends of ConductingEquipment with other classes.

Table 224 – Association ends of Core::ConductingEquipment with other classes

mult.	mult.	Class type name	Description
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.19 (ShortCircuit) MutualCoupling

This class represents the zero sequence line mutual coupling.

OCL constraint: First terminal of a mutual coupling must have a sequence number set

OCL constraint: Second terminal that of a mutual coupling must have a sequence number set

OCL constraint: The starting terminal for the calculation of distances along the first branch of the mutual coupling. Normally MutualCoupling would only be used for terminals of AC line segments. The terminals of a mutual coupling should point to different segments

OCL constraint: The starting terminal for the calculation of distances along the second branch of the mutual coupling. This should point to an ACLineSegment

Inheritance path: ->IdentifiedObject

Table 225 shows all attributes of MutualCoupling.

Table 225 – Attributes of Wires::MutualCoupling

Attribute name	mult.	Attribute type	Description
b0ch (ShortCircuit)	1..1	Susceptance	Zero sequence mutual coupling shunt (charging) susceptance, uniformly distributed, of the entire line section.
distance11 (ShortCircuit)	1..1	Length	Distance to the start of the coupled region from the first line's terminal having sequence number equal to 1.
distance12 (ShortCircuit)	1..1	Length	Distance to the end of the coupled region from the first line's terminal with sequence number equal to 1.
distance21 (ShortCircuit)	1..1	Length	Distance to the start of coupled region from the second line's terminal with sequence number equal to 1.
distance22 (ShortCircuit)	1..1	Length	Distance to the end of coupled region from the second line's terminal with sequence number equal to 1.
g0ch (ShortCircuit)	1..1	Conductance	Zero sequence mutual coupling shunt (charging) conductance, uniformly distributed, of the entire line section.
r0 (ShortCircuit)	1..1	Resistance	Zero sequence branch-to-branch mutual impedance coupling, resistance.
x0 (ShortCircuit)	1..1	Reactance	Zero sequence branch-to-branch mutual impedance coupling, reactance.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 226 shows all association ends of MutualCoupling with other classes.

Table 226 – Association ends of Wires::MutualCoupling with other classes

mult.	mult.	Class type name	Description
First_Terminal (ShortCircuit)	[1..1]	Terminal	The starting terminal for the calculation of distances along the first branch of the mutual coupling. Normally MutualCoupling would only be used for terminals of AC line segments. The first and second terminals of a mutual coupling should point to different AC line segments. Association Based On: Wires:: MutualCoupling.HasFirstMutualCoupling[0..*] ----- Core:: Terminal.First_Terminal[1]
Second_Terminal (ShortCircuit)	[1..1]	Terminal	The starting terminal for the calculation of distances along the second branch of the mutual coupling. Association Based On: Wires:: MutualCoupling.HasSecondMutualCoupling[0..*] ----- Core:: Terminal.Second_Terminal[1]

6.9.20 NonlinearShuntCompensator

A non linear shunt compensator has bank or section admittance values that differs.

Inheritance path: ->ShuntCompensator->RegulatingCondEq->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 227 shows all attributes of NonlinearShuntCompensator.

Table 227 – Attributes of Wires::NonlinearShuntCompensator

Attribute name	mult.	Attribute type	Description
aVRDelay	0..1	Seconds	see ShuntCompensator
grounded	0..1	Boolean	see ShuntCompensator
maximumSections	1..1	Integer	see ShuntCompensator
nomU	1..1	Voltage	see ShuntCompensator
normalSections	1..1	Integer	see ShuntCompensator
switchOnCount	0..1	Integer	see ShuntCompensator
switchOnDate	0..1	DateTime	see ShuntCompensator
voltageSensitivity	0..1	VoltagePerReactivePower	see ShuntCompensator
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 228 shows all association ends of RegulatingCondEq with other classes.

Table 228 – Association ends of Wires::RegulatingCondEq with other classes

mult.	mult.	Class type name	Description
RegulatingControl	0..1	RegulatingControl	Inherited Association Based On: Wires:: RegulatingCondEq.RegulatingCondEq[0..*] --- Wires:: RegulatingControl.RegulatingControl[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] --- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] --- Core:: Equipment.Equipments[0..*]

6.9.21 NonlinearShuntCompensatorPoint

A non linear shunt compensator bank or section admittance value.

Table 229 shows all attributes of NonlinearShuntCompensatorPoint.

Table 229 – Attributes of Wires::NonlinearShuntCompensatorPoint

Attribute name	mult.	Attribute type	Description
b	1..1	Susceptance	Positive sequence shunt (charging) susceptance per section
b0 (ShortCircuit)	1..1	Susceptance	Zero sequence shunt (charging) susceptance per section
g	1..1	Conductance	Positive sequence shunt (charging) conductance per section
g0 (ShortCircuit)	1..1	Conductance	Zero sequence shunt (charging) conductance per section
sectionNumber	1..1	Integer	The number of the section.

Table 230 shows all association ends of NonlinearShuntCompensatorPoint with other classes.

Table 230 – Association ends of Wires::NonlinearShuntCompensatorPoint with other classes

mult.	mult.	Class type name	Description
NonlinearShuntCompensator	[1..1]	NonlinearShuntCompensator	Non-linear shunt compensator owning this point. Association Based On: Wires:: NonlinearShuntCompensatorPoint.NonlinearShuntCompensatorPoints[1..*] ----- Wires:: NonlinearShuntCompensator.NonlinearShuntCompensator[1]

6.9.22 (ShortCircuit) PetersenCoil

A tunable impedance device normally used to offset line charging during single line faults in an ungrounded section of network.

Inheritance path: ->EarthFaultCompensator->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 231 shows all attributes of PetersenCoil.

Table 231 – Attributes of Wires::PetersenCoil

Attribute name	mult.	Attribute type	Description
mode (ShortCircuit)	1..1	PetersenCoilModeKind	The mode of operation of the Petersen coil.
nominalU (ShortCircuit)	1..1	Voltage	The nominal voltage for which the coil is designed.
offsetCurrent (ShortCircuit)	0..1	CurrentFlow	The offset current that the Petersen coil controller is operating from the resonant point. This is normally a fixed amount for which the controller is configured and could be positive or negative. Typically 0 to 60 Amperes depending on voltage and resonance conditions.
positionCurrent (ShortCircuit)	0..1	CurrentFlow	The control current used to control the Petersen coil also known as the position current. Typically in the range of 20-200mA.
xGroundMax (ShortCircuit)	1..1	Reactance	The maximum reactance.
xGroundMin (ShortCircuit)	1..1	Reactance	The minimum reactance.
xGroundNominal (ShortCircuit)	1..1	Reactance	The nominal reactance. This is the operating point (normally over compensation) that is defined based on the resonance point in the healthy network condition. The impedance is calculated based on nominal voltage divided by position current.
r	0..1	Resistance	see EarthFaultCompensator
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 232 shows all association ends of ConductingEquipment with other classes.

Table 232 – Association ends of Core::ConductingEquipment with other classes

mult.	mult.	Class type name	Description
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.23 PhaseTapChanger(Abstract)

A transformer phase shifting tap model that controls the phase angle difference across the power transformer and potentially the active power flow through the power transformer. This phase tap model may also impact the voltage magnitude.

Inheritance path: ->TapChanger->PowerSystemResource->IdentifiedObject

Table 233 shows all attributes of PhaseTapChanger.

Table 233 – Attributes of Wires::PhaseTapChanger

Attribute name	mult.	Attribute type	Description
highStep	1..1	Integer	see TapChanger
lowStep	1..1	Integer	see TapChanger
ltcFlag	1..1	Boolean	see TapChanger
neutralStep	1..1	Integer	see TapChanger
neutralU	1..1	Voltage	see TapChanger
normalStep	1..1	Integer	see TapChanger
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 234 shows all association ends of PhaseTapChanger with other classes.

Table 234 – Association ends of Wires::PhaseTapChanger with other classes

mult.	mult.	Class type name	Description
TransformerEnd	[1..1]	TransformerEnd	Phase tap changer associated with this transformer end. Association Based On: Wires:: TransformerEnd.TransformerEnd[1] ----- Wires:: PhaseTapChanger.PhaseTapChanger[0..1]
TapChangerControl	0..1	TapChangerControl	Inherited Association Based On: Wires:: TapChanger.TapChanger[0..*] ----- Wires:: TapChangerControl.TapChangerControl[0..1]

6.9.24 PhaseTapChangerAsymmetrical

Describes the tap model for an asymmetrical phase shifting transformer in which the difference voltage vector adds to the primary side voltage. The angle between the primary side voltage and the difference voltage is named the winding connection angle. The phase shift depends on both the difference voltage magnitude and the winding connection angle.

Inheritance path: ->PhaseTapChangerNonLinear->PhaseTapChanger->TapChanger->PowerSystemResource->IdentifiedObject

Table 235 shows all attributes of PhaseTapChangerAsymmetrical.

Table 235 – Attributes of Wires::PhaseTapChangerAsymmetrical

Attribute name	mult.	Attribute type	Description
windingConnectionAngle	1..1	AngleDegrees	The phase angle between the in-phase winding and the out-of -phase winding used for creating phase shift. The out-of-phase winding produces what is known as the difference voltage. Setting this angle to 90 degrees is not the same as a symmemtrical transformer.
voltageStepIncrement	1..1	PerCent	see PhaseTapChangerNonLinear
xMax	1..1	Reactance	see PhaseTapChangerNonLinear
xMin	1..1	Reactance	see PhaseTapChangerNonLinear
highStep	1..1	Integer	see TapChanger
lowStep	1..1	Integer	see TapChanger
ltcFlag	1..1	Boolean	see TapChanger
neutralStep	1..1	Integer	see TapChanger
neutralU	1..1	Voltage	see TapChanger
normalStep	1..1	Integer	see TapChanger
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 236 shows all association ends of PhaseTapChanger with other classes.

Table 236 – Association ends of Wires::PhaseTapChanger with other classes

mult.	mult.	Class type name	Description
TransformerEnd	1..1	TransformerEnd	Inherited Association Based On: Wires:: TransformerEnd.TransformerEnd[1] ----- Wires:: PhaseTapChanger.PhaseTapChanger[0..1]
TapChangerControl	0..1	TapChangerControl	Inherited Association Based On: Wires:: TapChanger.TapChanger[0..*] ----- Wires:: TapChangerControl.TapChangerControl[0..1]

6.9.25 PhaseTapChangerLinear

Describes a tap changer with a linear relation between the tap step and the phase angle difference across the transformer. This is a mathematical model that is an approximation of a real phase tap changer. The phase angle is computed as stepPhaseShitfIncrement times the tap position. The secondary side voltage magnitude is the same as at the primary side.

Inheritance path: ->PhaseTapChanger->TapChanger->PowerSystemResource->IdentifiedObject

Table 237 shows all attributes of PhaseTapChangerLinear.

Table 237 – Attributes of Wires::PhaseTapChangerLinear

Attribute name	mult.	Attribute type	Description
stepPhaseShiftIncrement	1..1	AngleDegrees	Phase shift per step position. A positive value indicates a positive phase shift from the winding where the tap is located to the other winding (for a two-winding transformer). The actual phase shift increment might be more accurately computed from the symmetrical or asymmetrical models or a tap step table lookup if those are available.
xMax	1..1	Reactance	The reactance depend on the tap position according to a "u" shaped curve. The maximum reactance (xMax) appear at the low and high tap positions.
xMin	1..1	Reactance	The reactance depend on the tap position according to a "u" shaped curve. The minimum reactance (xMin) appear at the mid tap position.
highStep	1..1	Integer	see TapChanger
lowStep	1..1	Integer	see TapChanger
ltcFlag	1..1	Boolean	see TapChanger
neutralStep	1..1	Integer	see TapChanger
neutralU	1..1	Voltage	see TapChanger
normalStep	1..1	Integer	see TapChanger
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 238 shows all association ends of PhaseTapChanger with other classes.

Table 238 – Association ends of Wires::PhaseTapChanger with other classes

mult.	mult.	Class type name	Description
TransformerEnd	1..1	TransformerEnd	Inherited Association Based On: Wires:: TransformerEnd.TransformerEnd[1] ----- Wires:: PhaseTapChanger.PhaseTapChanger[0..1]
TapChangerControl	0..1	TapChangerControl	Inherited Association Based On: Wires:: TapChanger.TapChanger[0..*] ----- Wires:: TapChangerControl.TapChangerControl[0..1]

6.9.26 PhaseTapChangerNonLinear(Abstract)

The non-linear phase tap changer describes the non-linear behavior of a phase tap changer. This is a base class for the symmetrical and asymmetrical phase tap changer models. The details of these models can be found in IEC 61970-301.

Inheritance path: ->PhaseTapChanger->TapChanger->PowerSystemResource->IdentifiedObject

Table 239 shows all attributes of PhaseTapChangerNonLinear.

Table 239 – Attributes of Wires::PhaseTapChangerNonLinear

Attribute name	mult.	Attribute type	Description
voltageStepIncrement	1..1	PerCent	The voltage step increment on the out of phase winding specified in percent of nominal voltage of the transformer end.
xMax	1..1	Reactance	The reactance depend on the tap position according to a "u" shaped curve. The maximum reactance (xMax) appear at the low and high tap positions.
xMin	1..1	Reactance	The reactance depend on the tap position according to a "u" shaped curve. The minimum reactance (xMin) appear at the mid tap position.
highStep	1..1	Integer	see TapChanger
lowStep	1..1	Integer	see TapChanger
ltcFlag	1..1	Boolean	see TapChanger
neutralStep	1..1	Integer	see TapChanger
neutralU	1..1	Voltage	see TapChanger
normalStep	1..1	Integer	see TapChanger
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 240 shows all association ends of PhaseTapChanger with other classes.

Table 240 – Association ends of Wires::PhaseTapChanger with other classes

mult.	mult.	Class type name	Description
TransformerEnd	1..1	TransformerEnd	Inherited Association Based On: Wires:: TransformerEnd.TransformerEnd[1] ----- Wires:: PhaseTapChanger.PhaseTapChanger[0..1]
TapChangerControl	0..1	TapChangerControl	Inherited Association Based On: Wires:: TapChanger.TapChanger[0..*] ----- Wires:: TapChangerControl.TapChangerControl[0..1]

6.9.27 PhaseTapChangerSymmetrical

Describes a symmetrical phase shifting transformer tap model in which the secondary side voltage magnitude is the same as at the primary side. The difference voltage magnitude is the base in an equal-sided triangle where the sides corresponds to the primary and secondary voltages. The phase angle difference corresponds to the top angle and can be expressed as twice the arctangent of half the total difference voltage.

Inheritance path: ->PhaseTapChangerNonLinear->PhaseTapChanger->TapChanger->PowerSystemResource->IdentifiedObject

Table 241 shows all attributes of PhaseTapChangerSymmetrical.

Table 241 – Attributes of Wires::PhaseTapChangerSymmetrical

Attribute name	mult.	Attribute type	Description
voltageStepIncrement	1..1	PerCent	see PhaseTapChangerNonLinear
xMax	1..1	Reactance	see PhaseTapChangerNonLinear
xMin	1..1	Reactance	see PhaseTapChangerNonLinear
highStep	1..1	Integer	see TapChanger
lowStep	1..1	Integer	see TapChanger
ltcFlag	1..1	Boolean	see TapChanger
neutralStep	1..1	Integer	see TapChanger
neutralU	1..1	Voltage	see TapChanger
normalStep	1..1	Integer	see TapChanger
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 242 shows all association ends of PhaseTapChanger with other classes.

Table 242 – Association ends of Wires::PhaseTapChanger with other classes

mult.	mult.	Class type name	Description
TransformerEnd	1..1	TransformerEnd	Inherited Association Based On: Wires:: TransformerEnd.TransformerEnd[1] ----- Wires:: PhaseTapChanger.PhaseTapChanger[0..1]
TapChangerControl	0..1	TapChangerControl	Inherited Association Based On: Wires:: TapChanger.TapChanger[0..*] ----- Wires:: TapChangerControl.TapChangerControl[0..1]

6.9.28 PhaseTapChangerTable

Describes a tabular curve for how the phase angle difference and impedance varies with the tap step.

Inheritance path: ->IdentifiedObject

Table 243 shows all attributes of PhaseTapChangerTable.

Table 243 – Attributes of Wires::PhaseTapChangerTable

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.9.29 PhaseTapChangerTablePoint

Describes each tap step in the phase tap changer tabular curve.

Inheritance path: ->TapChangerTablePoint

Table 244 shows all attributes of PhaseTapChangerTablePoint.

Table 244 – Attributes of Wires::PhaseTapChangerTablePoint

Attribute name	mult.	Attribute type	Description
angle	1..1	AngleDegrees	The angle difference in degrees.
b	0..1	PerCent	see TapChangerTablePoint
g	0..1	PerCent	see TapChangerTablePoint
r	0..1	PerCent	see TapChangerTablePoint
ratio	0..1	Simple_Float	see TapChangerTablePoint
step	1..1	Integer	see TapChangerTablePoint
x	0..1	PerCent	see TapChangerTablePoint

Table 245 shows all association ends of PhaseTapChangerTablePoint with other classes.

Table 245 – Association ends of Wires::PhaseTapChangerTablePoint with other classes

mult.	mult.	Class type name	Description
PhaseTapChangerTable	[1..1]	PhaseTapChangerTable	The table of this point. Association Based On: Wires:: PhaseTapChangerTablePoint.PhaseTapChangerTablePoint[1..*] ----- Wires:: PhaseTapChangerTable.PhaseTapChangerTable[1]

6.9.30 PhaseTapChangerTabular

Inheritance path: ->PhaseTapChanger->TapChanger->PowerSystemResource->IdentifiedObject

Table 246 shows all attributes of PhaseTapChangerTabular.

Table 246 – Attributes of Wires::PhaseTapChangerTabular

Attribute name	mult.	Attribute type	Description
highStep	1..1	Integer	see TapChanger
lowStep	1..1	Integer	see TapChanger
ltcFlag	1..1	Boolean	see TapChanger
neutralStep	1..1	Integer	see TapChanger
neutralU	1..1	Voltage	see TapChanger
normalStep	1..1	Integer	see TapChanger
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 247 shows all association ends of PhaseTapChangerTabular with other classes.

Table 247 – Association ends of Wires::PhaseTapChangerTabular with other classes

mult.	mult.	Class type name	Description
PhaseTapChangerTable	[0..1]	PhaseTapChangerTable	The phase tap changer table for this phase tap changer. Association Based On: Wires:: PhaseTapChangerTable.PhaseTapChangerTable[0..1] ---- Wires:: PhaseTapChangerTabular.PhaseTapChangerTabular[0..*]
TransformerEnd	1..1	TransformerEnd	Inherited Association Based On: Wires:: TransformerEnd.TransformerEnd[1] ----- Wires:: PhaseTapChanger.PhaseTapChanger[0..1]
TapChangerControl	0..1	TapChangerControl	Inherited Association Based On: Wires:: TapChanger.TapChanger[0..*] ----- Wires:: TapChangerControl.TapChangerControl[0..1]

6.9.31 PowerTransformer

An electrical device consisting of two or more coupled windings, with or without a magnetic core, for introducing mutual coupling between electric circuits. Transformers can be used to control voltage and phase shift (active power flow). A power transformer may be composed of separate transformer tanks that need not be identical.

A power transformer can be modeled with or without tanks and is intended for use in both balanced and unbalanced representations. A power transformer typically has two terminals, but may have one (grounding), three or more terminals. The inherited association ConductingEquipment.BaseVoltage should not be used. The association from TransformerEnd to BaseVoltage should be used instead.

A PowerTransformer is contained in one Substation but it can connect a Terminal to another Substation.

Association between PowerTransformer and BaseVoltage is not exchanged.

PowerTransformer now inherits from ConductingEquipment instead of Equipment. The terminals that were formerly attached to TransformerWinding are now attached to PowerTransformer. The Terminal.sequenceNumber distinquishes the terminals much as previously done by TransformerWinding.windingType:WindingType. A PowerTransformer may be balanced or unbalanced and may optionally model unbalanced tank level detail. The Terminal to PowerTransformerEnd association in addition to the Terminal to ConductingEquipment (to PowerTransformer instances) association is used. There is one terminal per PoweTransformerEnd, i.e. for a 2-winding transformer in total two terminals. Each terminal would have an association to the same PowerTransformer instance and one association to the individual PowerTransformerEnd instances. TransformerEnd.endNumber attribute is used to define high and lower sides of the PowerTransformer.

A PowerTransformer could be associated to 2 or 3 Terminals.

A PowerTransformer can be either two winding or three winding. – A two winding transformer has two PowerTransformerEnds – A three winding transformer has three PowerTransformerEnds.

Naming Convention and main containership hierarchy is:
GeographicalRegion/SubGeographicalRegion/Substation

Inheritance path: ->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 248 shows all attributes of PowerTransformer.

Table 248 – Attributes of Wires::PowerTransformer

Attribute name	mult.	Attribute type	Description
beforeShCircuitHighestOperatingCurrent (ShortCircuit)	0..1	CurrentFlow	The highest operating current (Ib in the IEC 60909-0) before short circuit (depends on network configuration and relevant reliability philosophy). It is used for calculation of the impedance correction factor KT defined in IEC 60909-0.
beforeShCircuitHighestOperatingVoltage (ShortCircuit)	0..1	Voltage	The highest operating voltage (Ub in the IEC 60909-0) before short circuit. It is used for calculation of the impedance correction factor KT defined in IEC 60909-0:2016. This is worst case voltage on the low side winding (3.7.1). Used to define operating conditions.
beforeShortCircuitAnglePf (ShortCircuit)	0..1	AngleDegrees	The angle of power factor before short circuit (phib in the IEC 60909-0). It is used for calculation of the impedance correction factor KT defined in IEC 60909-0. This is the worst case power factor. Used to define operating conditions.
highSideMinOperatingU (ShortCircuit)	0..1	Voltage	The minimum operating voltage (uQmin in the IEC 60909-0) at the high voltage side (Q side) of the unit transformer of the power station unit. A value well established from long-term operating experience of the system. It is used for calculation of the impedance correction factor KG defined in IEC 60909-0.
isPartOfGeneratorUnit (ShortCircuit)	1..1	Boolean	Indicates whether the machine is part of a power station unit. Used for short circuit data exchange according to IEC 60909.
operationalValuesConsidered (ShortCircuit)	0..1	Boolean	It is used to define if the data (other attributes related to short circuit data exchange) defines long term operational conditions or not. Used for short circuit data exchange according to IEC 60909.
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 249 shows all association ends of ConductingEquipment with other classes.

Table 249 – Association ends of Core::ConductingEquipment with other classes

mult.	mult.	Class type name	Description
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.32 PowerTransformerEnd

A PowerTransformerEnd is associated with each Terminal of a PowerTransformer.

The impedance values r, r0, x, and x0 of a PowerTransformerEnd represents a star equivalent as follows

1) for a two Terminal PowerTransformer the high voltage PowerTransformerEnd has non zero values on r, r0, x, and x0 while the low voltage PowerTransformerEnd has zero values for r, r0, x, and x0.
2) for a three Terminal PowerTransformer the three PowerTransformerEnds represents a star equivalent with each leg in the star represented by r, r0, x, and x0 values.
3) for a PowerTransformer with more than three Terminals the PowerTransformerEnd impedance values cannot be used. Instead use the TransformerMeshImpedance or split the transformer into multiple PowerTransformers.
   – Each PowerTransformerEnd must be contained by a PowerTransformer. Because a PowerTransformerEnd (or any other object) can not be contained by more than one parent, a PowerTransformerEnd can not have an association to an EquipmentContainer (Substation, VoltageLevel, etc.).

Inheritance path: ->TransformerEnd->IdentifiedObject

Table 250 shows all attributes of PowerTransformerEnd.

Table 250 – Attributes of Wires::PowerTransformerEnd

Attribute name	mult.	Attribute type	Description
b	1..1	Susceptance	Magnetizing branch susceptance (B mag). The value can be positive or negative.
connectionKind	0..1	WindingConnection	Kind of connection.
b0 (ShortCircuit)	1..1	Susceptance	Zero sequence magnetizing branch susceptance.
phaseAngleClock (ShortCircuit)	1..1	Integer	Terminal voltage phase angle displacement where 360 degrees are represented with clock hours. The valid values are 0 to 11. For example, for the secondary side end of a transformer with vector group code of 'Dyn11', specify the connection kind as wye with neutral and specify the phase angle of the clock as 11. The clock value of the transformer end number specified as 1, is assumed to be zero. Note the transformer end number is not assumed to be the same as the terminal sequence number.
ratedS	0..1	ApparentPower	Normal apparent power rating. The attribute shall be a positive value. For a two-winding transformer the values for the high and low voltage sides shall be identical.
g	0..1	Conductance	Magnetizing branch conductance.
ratedU	1..1	Voltage	Rated voltage: phase-phase for three-phase windings, and either phase-phase or phase-neutral for single-phase windings. A high voltage side, as given by TransformerEnd.endNumber, shall have a ratedU that is greater or equal than ratedU for the lower voltage sides.
g0 (ShortCircuit)	1..1	Conductance	Zero sequence magnetizing branch conductance (star-model).
r	1..1	Resistance	Resistance (star-model) of the transformer end. The attribute shall be equal or greater than zero for non-equivalent transformers.
r0 (ShortCircuit)	1..1	Resistance	Zero sequence series resistance (star-model) of the transformer end.
x	1..1	Reactance	Positive sequence series reactance (star-model) of the transformer end.
x0 (ShortCircuit)	1..1	Reactance	Zero sequence series reactance of the transformer end.
rground (ShortCircuit)	0..1	Resistance	see TransformerEnd
endNumber	1..1	Integer	see TransformerEnd
grounded (ShortCircuit)	1..1	Boolean	see TransformerEnd
xground (ShortCircuit)	0..1	Reactance	see TransformerEnd
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 251 shows all association ends of PowerTransformerEnd with other classes.

Table 251 – Association ends of Wires::PowerTransformerEnd with other classes

mult.	mult.	Class type name	Description
PowerTransformer	[1..1]	PowerTransformer	The ends of this power transformer. Association Based On: Wires:: PowerTransformer.PowerTransformer[0..1] ----- Wires:: PowerTransformerEnd.PowerTransformerEnd[0..*]
Terminal	1..1	Terminal	Inherited Association Based On: Wires:: TransformerEnd.TransformerEnd[0..*] ----- Core:: Terminal.Terminal[0..1]
BaseVoltage	1..1	BaseVoltage	Inherited Association Based On: Wires:: TransformerEnd.TransformerEnds[0..*] ----- Core:: BaseVoltage.BaseVoltage[0..1]

6.9.33 ProtectedSwitch(Abstract)

A ProtectedSwitch is a switching device that can be operated by ProtectionEquipment.

Inheritance path: ->Switch->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 252 shows all attributes of ProtectedSwitch.

Table 252 – Attributes of Wires::ProtectedSwitch

Attribute name	mult.	Attribute type	Description
normalOpen	1..1	Boolean	see Switch
ratedCurrent	0..1	CurrentFlow	see Switch
retained	1..1	Boolean	see Switch
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 253 shows all association ends of ConductingEquipment with other classes.

Table 253 – Association ends of Core::ConductingEquipment with other classes

mult.	mult.	Class type name	Description
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.34 RatioTapChanger

A tap changer that changes the voltage ratio impacting the voltage magnitude but not the phase angle across the transformer

- The attribute TapChanger.ltcflag specifies whether or not a TapChanger has load tap changing capabilities. If the ltcFlag is true, the attribute "stepVoltageIncrement" is required.

Inheritance path: ->TapChanger->PowerSystemResource->IdentifiedObject

Table 254 shows all attributes of RatioTapChanger.

Table 254 – Attributes of Wires::RatioTapChanger

Attribute name	mult.	Attribute type	Description
tculControlMode	1..1	TransformerControlMode	Specifies the regulation control mode (voltage or reactive) of the RatioTapChanger.
stepVoltageIncrement	1..1	PerCent	Tap step increment, in per cent of nominal voltage, per step position.
highStep	1..1	Integer	see TapChanger
lowStep	1..1	Integer	see TapChanger
ltcFlag	1..1	Boolean	see TapChanger
neutralStep	1..1	Integer	see TapChanger
neutralU	1..1	Voltage	see TapChanger
normalStep	1..1	Integer	see TapChanger
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 255 shows all association ends of RatioTapChanger with other classes.

Table 255 – Association ends of Wires::RatioTapChanger with other classes

mult.	mult.	Class type name	Description
RatioTapChangerTable	[0..1]	RatioTapChangerTable	The ratio tap changer of this tap ratio table. Association Based On: Wires:: RatioTapChanger.RatioTapChanger[0..*] ----- Wires:: RatioTapChangerTable.RatioTapChangerTable[0..1]
TransformerEnd	[1..1]	TransformerEnd	Ratio tap changer associated with this transformer end. Association Based On: Wires:: TransformerEnd.TransformerEnd[1] ----- Wires:: RatioTapChanger.RatioTapChanger[0..1]
TapChangerControl	0..1	TapChangerControl	Inherited Association Based On: Wires:: TapChanger.TapChanger[0..*] ----- Wires:: TapChangerControl.TapChangerControl[0..1]

6.9.35 RatioTapChangerTable

Describes a curve for how the voltage magnitude and impedance varies with the tap step.

Inheritance path: ->IdentifiedObject

Table 256 shows all attributes of RatioTapChangerTable.

Table 256 – Attributes of Wires::RatioTapChangerTable

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.9.36 RatioTapChangerTablePoint

Describes each tap step in the ratio tap changer tabular curve.

Inheritance path: ->TapChangerTablePoint

Table 257 shows all attributes of RatioTapChangerTablePoint.

Table 257 – Attributes of Wires::RatioTapChangerTablePoint

Attribute name	mult.	Attribute type	Description
b	0..1	PerCent	see TapChangerTablePoint
g	0..1	PerCent	see TapChangerTablePoint
r	0..1	PerCent	see TapChangerTablePoint
ratio	0..1	Simple_Float	see TapChangerTablePoint
step	1..1	Integer	see TapChangerTablePoint
x	0..1	PerCent	see TapChangerTablePoint

Table 258 shows all association ends of RatioTapChangerTablePoint with other classes.

Table 258 – Association ends of Wires::RatioTapChangerTablePoint with other classes

mult.	mult.	Class type name	Description
RatioTapChangerTable	[1..1]	RatioTapChangerTable	Points of this table. Association Based On: Wires:: RatioTapChangerTablePoint.RatioTapChangerTablePoint[1..*] ----- Wires:: RatioTapChangerTable.RatioTapChangerTable[1]

6.9.37 ReactiveCapabilityCurve

Reactive power rating envelope versus the synchronous machine's active power, in both the generating and motoring modes. For each active power value there is a corresponding high and low reactive power limit value. Typically there will be a separate curve for each coolant condition, such as hydrogen pressure. The Y1 axis values represent reactive minimum and the Y2 axis values represent reactive maximum.

- ReactiveCapabilityCurves are not required if the reactive power limits of the SynchronousMachine do not vary with real power output.
- By convention, the Y1 axis values represent reactive minimum and the Y2 axis values represent reactive maximum.
- Because the x value will always be specified in MW and the y values will always be specified in MVAr, the xMultiplier, y1Multiplier, and y2Multiplier attributes do not need to be supplied.

Inheritance path: ->Curve->IdentifiedObject

Table 259 shows all attributes of ReactiveCapabilityCurve.

Table 259 – Attributes of Wires::ReactiveCapabilityCurve

Attribute name	mult.	Attribute type	Description
curveStyle	1..1	CurveStyle	see Curve
xUnit	1..1	UnitSymbol	see Curve
y1Unit	1..1	UnitSymbol	see Curve
y2Unit	0..1	UnitSymbol	see Curve
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.9.38 RegulatingCondEq(Abstract)

A type of conducting equipment that can regulate a quantity (i.e. voltage or flow) at a specific point in the network.

Inheritance path: ->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 260 shows all attributes of RegulatingCondEq.

Table 260 – Attributes of Wires::RegulatingCondEq

Attribute name	mult.	Attribute type	Description
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 261 shows all association ends of RegulatingCondEq with other classes.

Table 261 – Association ends of Wires::RegulatingCondEq with other classes

mult.	mult.	Class type name	Description
RegulatingControl	[0..1]	RegulatingControl	The regulating control scheme in which this equipment participates. Association Based On: Wires:: RegulatingCondEq.RegulatingCondEq[0..*] ----- Wires:: RegulatingControl.RegulatingControl[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.39 RegulatingControl

Specifies a set of equipment that works together to control a power system quantity such as voltage or flow. Remote bus voltage control is possible by specifying the controlled terminal located at some place remote from the controlling equipment. In case multiple equipment, possibly of different types, control same terminal there must be only one RegulatingControl at that terminal. The most specific subtype of RegulatingControl shall be used in case such equipment participate in the control, e.g. TapChangerControl for tap changers. For flow control load sign convention is used, i.e. positive sign means flow out from a TopologicalNode (bus) into the conducting equipment.

RegulatingControl targetrange and targetvalue are required if a RegulationSchedule is not provided.

Inheritance path: ->PowerSystemResource->IdentifiedObject

Table 262 shows all attributes of RegulatingControl.

Table 262 – Attributes of Wires::RegulatingControl

Attribute name	mult.	Attribute type	Description
mode	1..1	RegulatingControlModeKind	The regulating control mode presently available. This specification allows for determining the kind of regulation without need for obtaining the units from a schedule.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 263 shows all association ends of RegulatingControl with other classes.

Table 263 – Association ends of Wires::RegulatingControl with other classes

mult.	mult.	Class type name	Description
Terminal	[1..1]	Terminal	The controls regulating this terminal. Association Based On: Core:: Terminal.Terminal[0..1] ----- Wires:: RegulatingControl.RegulatingControl[0..*]

6.9.40 (Operation) RegulationSchedule

A pre-established pattern over time for a controlled variable, e.g., busbar voltage.

- By convention, ”value1” represents the target voltage or real power. “value2” is the deviation. A value1 of 100 and value2 of 1 means regulating to 100KV plus or minus 1KV. The range would be from 99 KV to 101 KV. Because the regulation values will be specified in either kV for voltage or MW for real power, the value1Multiplier and value2Multiplier attributes do not need to be specified.

Inheritance path: ->SeasonDayTypeSchedule->RegularIntervalSchedule->BasicIntervalSchedule->IdentifiedObject

Table 264 shows all attributes of RegulationSchedule.

Table 264 – Attributes of Wires::RegulationSchedule

Attribute name	mult.	Attribute type	Description
timeStep	1..1	Seconds	see RegularIntervalSchedule
endTime	1..1	DateTime	see RegularIntervalSchedule
startTime	1..1	DateTime	see BasicIntervalSchedule
value1Unit	1..1	UnitSymbol	see BasicIntervalSchedule
value2Unit	0..1	UnitSymbol	see BasicIntervalSchedule
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 265 shows all association ends of RegulationSchedule with other classes.

Table 265 – Association ends of Wires::RegulationSchedule with other classes

mult.	mult.	Class type name	Description
RegulatingControl (Operation)	[1..1]	RegulatingControl	Regulating controls that have this Schedule. Association Based On: Wires:: RegulationSchedule.RegulationSchedule[0..*] ----- Wires:: RegulatingControl.RegulatingControl[1]
Season (Operation)	1..1	Season	Inherited Association Based On: LoadModel:: Season.Season[0..1] ----- LoadModel:: SeasonDayTypeSchedule.SeasonDayTypeSchedules[0..*]
DayType (Operation)	1..1	DayType	Inherited Association Based On: LoadModel:: DayType.DayType[0..1] ----- LoadModel:: SeasonDayTypeSchedule.SeasonDayTypeSchedules[0..*]

6.9.41 RotatingMachine(Abstract)

A rotating machine which may be used as a generator or motor.

RotatingMachine.ratedS is required in case of dynamic data exchange.

If one RotatingMachine is associated with one GeneratingUnit the flag Equipment.aggregate should be consistent in case it is provided at both RotatingMachine and GeneratingUnit.

Inheritance path: ->RegulatingCondEq->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 266 shows all attributes of RotatingMachine.

Table 266 – Attributes of Wires::RotatingMachine

Attribute name	mult.	Attribute type	Description
ratedPowerFactor	0..1	Simple_Float	Power factor (nameplate data). It is primarily used for short circuit data exchange according to IEC 60909.
ratedS	0..1	ApparentPower	Nameplate apparent power rating for the unit. The attribute shall have a positive value.
ratedU	0..1	Voltage	Rated voltage (nameplate data, Ur in IEC 60909-0). It is primarily used for short circuit data exchange according to IEC 60909.
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 267 shows all association ends of RotatingMachine with other classes.

Table 267 – Association ends of Wires::RotatingMachine with other classes

mult.	mult.	Class type name	Description
GeneratingUnit	[0..1]	GeneratingUnit	A synchronous machine may operate as a generator and as such becomes a member of a generating unit. Association Based On: Production:: GeneratingUnit.GeneratingUnit[0..1] ----- Wires:: RotatingMachine.RotatingMachine[1..*]
RegulatingControl	0..1	RegulatingControl	Inherited Association Based On: Wires:: RegulatingCondEq.RegulatingCondEq[0..*] ----- Wires:: RegulatingControl.RegulatingControl[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.42 SeriesCompensator

A Series Compensator is a series capacitor or reactor or an AC transmission line without charging susceptance. It is a two terminal device.

Each SeriesCompensator is required to have an association to a BaseVoltage.

The positive sequence resistance (r) and reactance (x) are the same as negative sequence. Therefore negative sequence data is not modeled.

Inheritance path: ->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 268 shows all attributes of SeriesCompensator.

Table 268 – Attributes of Wires::SeriesCompensator

Attribute name	mult.	Attribute type	Description
r	1..1	Resistance	Positive sequence resistance.
r0 (ShortCircuit)	1..1	Resistance	Zero sequence resistance.
x	1..1	Reactance	Positive sequence reactance.
x0 (ShortCircuit)	1..1	Reactance	Zero sequence reactance.
varistorPresent	1..1	Boolean	Describe if a metal oxide varistor (mov) for over voltage protection is configured at the series compensator.
varistorRatedCurrent	1..1	CurrentFlow	The maximum current the varistor is designed to handle at specified duration.
varistorVoltageThreshold	1..1	Voltage	The dc voltage at which the varistor start conducting.
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 269 shows all association ends of ConductingEquipment with other classes.

Table 269 – Association ends of Core::ConductingEquipment with other classes

mult.	mult.	Class type name	Description
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] --- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] --- Core:: Equipment.Equipments[0..*]

6.9.43 ShuntCompensator(Abstract)

A shunt capacitor or reactor or switchable bank of shunt capacitors or reactors. A section of a shunt compensator is an individual capacitor or reactor. A negative value for reactivePerSection indicates that the compensator is a reactor. ShuntCompensator is a single terminal device. Ground is implied.

Inheritance path: ->RegulatingCondEq->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 270 shows all attributes of ShuntCompensator.

Table 270 – Attributes of Wires::ShuntCompensator

Attribute name	mult.	Attribute type	Description
aVRDelay	0..1	Seconds	Time delay required for the device to be connected or disconnected by automatic voltage regulation (AVR).
grounded	0..1	Boolean	Used for Yn and Zn connections. True if the neutral is solidly grounded.
maximumSections	1..1	Integer	The maximum number of sections that may be switched in.
nomU	1..1	Voltage	The voltage at which the nominal reactive power may be calculated. This should normally be within 10% of the voltage at which the capacitor is connected to the network.
normalSections	1..1	Integer	The normal number of sections switched in.
switchOnCount	0..1	Integer	The switch on count since the capacitor count was last reset or initialized.
switchOnDate	0..1	DateTime	The date and time when the capacitor bank was last switched on.
voltageSensitivity	0..1	VoltagePerReactivePower	Voltage sensitivity required for the device to regulate the bus voltage, in voltage/reactive power.
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 271 shows all association ends of RegulatingCondEq with other classes.

Table 271 – Association ends of Wires::RegulatingCondEq with other classes

mult.	mult.	Class type name	Description
RegulatingControl	0..1	RegulatingControl	Inherited Association Based On: Wires:: RegulatingCondEq.RegulatingCondEq[0..*] ----- Wires:: RegulatingControl.RegulatingControl[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.44 StaticVarCompensator

A facility for providing variable and controllable shunt reactive power. The SVC typically consists of a stepdown transformer, filter, thyristor-controlled reactor, and thyristor-switched capacitor arms. The SVC may operate in fixed MVar output mode or in voltage control mode. When in voltage control mode, the output of the SVC will be proportional to the deviation of voltage at the controlled bus from the voltage setpoint. The SVC characteristic slope defines the proportion. If the voltage at the controlled bus is equal to the voltage setpoint, the SVC MVar output is zero.

Naming Convention and main containership hierarchy is:

GeographicalRegion/SubGeographicalRegion/Substation/VoltageLevel

- The value of the “inductiveRating” attribute must always be negative. – The value of the “capactiveRating” attribute must always be positive.
- It is required to use RegulatingControl for SVC.

Inheritance path: ->RegulatingCondEq->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 272 shows all attributes of StaticVarCompensator.

Table 272 – Attributes of Wires::StaticVarCompensator

Attribute name	mult.	Attribute type	Description
capacitiveRating	1..1	Reactance	Maximum available capacitive reactance.
inductiveRating	1..1	Reactance	Maximum available inductive reactance.
slope	1..1	VoltagePerReactivePower	The characteristics slope of an SVC defines how the reactive power output changes in proportion to the difference between the regulated bus voltage and the voltage setpoint.
sVCControlMode	1..1	SVCControlMode	SVC control mode.
voltageSetPoint	1..1	Voltage	The reactive power output of the SVC is proportional to the difference between the voltage at the regulated bus and the voltage setpoint. When the regulated bus voltage is equal to the voltage setpoint, the reactive power output is zero.
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 273 shows all association ends of RegulatingCondEq with other classes.

Table 273 – Association ends of Wires::RegulatingCondEq with other classes

mult.	mult.	Class type name	Description
RegulatingControl	0..1	RegulatingControl	Inherited Association Based On: Wires:: RegulatingCondEq.RegulatingCondEq[0..*] ----- Wires:: RegulatingControl.RegulatingControl[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.45 Switch

A generic device designed to close, or open, or both, one or more electric circuits. All switches are two terminal devices including grounding switches.

For switching Devices, Naming Convention and main containership hierarchy is: GeographicalRegion/SubGeographicalRegion/Substation/VoltageLevel or GeographicalRegion/SubGeographicalRegion/SubstationàVoltageLevel/Bay

Instance data to be exchanged MUST make use of the most detailed class possible. The class Switch should only be used if the information to determine the more detailed class (Breaker, Disconnector, etc.) is not available.

Inheritance path: ->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 274 shows all attributes of Switch.

Table 274 – Attributes of Wires::Switch

Attribute name	mult.	Attribute type	Description
normalOpen	1..1	Boolean	The attribute is used in cases when no Measurement for the status value is present. If the Switch has a status measurement the Discrete.normalValue is expected to match with the Switch.normalOpen.
ratedCurrent	0..1	CurrentFlow	The maximum continuous current carrying capacity in amps governed by the device material and construction.
retained	1..1	Boolean	Branch is retained in a bus branch model. The flow through retained switches will normally be calculated in power flow.
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 275 shows all association ends of ConductingEquipment with other classes.

Table 275 – Association ends of Core::ConductingEquipment with other classes

mult.	mult.	Class type name	Description
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.46 (Operation) SwitchSchedule

A schedule of switch positions. If RegularTimePoint.value1 is 0, the switch is open. If 1, the switch is closed.

Inheritance path: ->SeasonDayTypeSchedule->RegularIntervalSchedule->BasicIntervalSchedule->IdentifiedObject

Table 276 shows all attributes of SwitchSchedule.

Table 276 – Attributes of Wires::SwitchSchedule

Attribute name	mult.	Attribute type	Description
timeStep	1..1	Seconds	see RegularIntervalSchedule
endTime	1..1	DateTime	see RegularIntervalSchedule
startTime	1..1	DateTime	see BasicIntervalSchedule
value1Unit	1..1	UnitSymbol	see BasicIntervalSchedule
value2Unit	0..1	UnitSymbol	see BasicIntervalSchedule
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 277 shows all association ends of SwitchSchedule with other classes.

Table 277 – Association ends of Wires::SwitchSchedule with other classes

mult.	mult.	Class type name	Description
Switch (Operation)	[1..1]	Switch	A Switch can be associated with SwitchSchedules. Association Based On: Wires:: Switch.Switch[1] ----- Wires:: SwitchSchedule.SwitchSchedules[0..*]
Season (Operation)	1..1	Season	Inherited Association Based On: LoadModel:: Season.Season[0..1] ----- LoadModel:: SeasonDayTypeSchedule.SeasonDayTypeSchedules[0..*]
DayType (Operation)	1..1	DayType	Inherited Association Based On: LoadModel:: DayType.DayType[0..1] ----- LoadModel:: SeasonDayTypeSchedule.SeasonDayTypeSchedules[0..*]

6.9.47 SynchronousMachine

An electromechanical device that operates with shaft rotating synchronously with the network. It is a single machine operating either as a generator or synchronous condenser or pump.

OCL constraint:If a synchronous condenser is being modelled so that there is no capability for real power output, the SynchronousMachine is not required to be associated with a GeneratingUnit. In this case, the type and operatingMode attributes must both be set to "cond

The attribute satDirectTransX is not required for short circuit data exchange according to IEC 60909.

Naming Convention and main containership hierarchy is:
GeographicalRegion/SubGeographicalRegion/Substation/VoltageLevel

- If SynchronousMachine.earthing is true, then SynchronousMachine.earthingStarPointR and SynchronousMachine.earthingStarPointX are required.

- If a SynchronousMachine is not associated with a ReactiveCapabilityCurve, then the minQ and maxQ attributes will be used.
- If a ReactiveCapabilityCurve is supplied, then the minQ and maxQ attributes are not required. – If a synchronous condenser is being modeled so that there is no capability for real power output, the SynchronousMachine is not required to be associated with a GeneratingUnit. In this case, the type and operatingMode attributes must both be set to “condenser”.

Inheritance path: ->RotatingMachine->RegulatingCondEq->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 278 shows all attributes of SynchronousMachine.

Table 278 – Attributes of Wires::SynchronousMachine

Attribute name	mult.	Attribute type	Description
earthing (ShortCircuit)	1..1	Boolean	Indicates whether or not the generator is earthed. Used for short circuit data exchange according to IEC 60909
earthingStarPointR (ShortCircuit)	0..1	Resistance	Generator star point earthing resistance (Re). Used for short circuit data exchange according to IEC 60909
earthingStarPointX (ShortCircuit)	0..1	Reactance	Generator star point earthing reactance (Xe). Used for short circuit data exchange according to IEC 60909
ikk (ShortCircuit)	0..1	CurrentFlow	Steady-state short-circuit current (in A for the profile) of generator with compound excitation during 3-phase short circuit. - Ikk=0: Generator with no compound excitation. - Ikk?0: Generator with compound excitation. Ikk is used to calculate the minimum steady-state short-circuit current for generators with compound excitation (4.6.1.2 of IEC 60909-0:2016) Used only for single fed short circuit on a generator. (4.3.4.2. of IEC 60909-0:2016)
maxQ	0..1	ReactivePower	Maximum reactive power limit. This is the maximum (nameplate) limit for the unit.
minQ	0..1	ReactivePower	Minimum reactive power limit for the unit.
mu (ShortCircuit)	0..1	Simple_Float	Factor to calculate the breaking current (4.5.2.1 of IEC 60909-0:2016). Used only for single fed short circuit on a generator (4.3.4.2. of IEC 60909-0:2016).
qPercent	0..1	PerCent	Percent of the coordinated reactive control that comes from this machine.
r0 (ShortCircuit)	1..1	PU	Zero sequence resistance of the synchronous machine.
r2 (ShortCircuit)	1..1	PU	Negative sequence resistance.
satDirectSubtransX (ShortCircuit)	1..1	PU	Direct-axis subtransient reactance saturated, also known as Xd"sat.

Attribute name	mult.	Attribute type	Description
satDirectSyncX (ShortCircuit)	0..1	PU	Direct-axes saturated synchronous reactance (xdsat); reciprocal of short-circuit ration. Used for short circuit data exchange, only for single fed short circuit on a generator. (4.3.4.2. of IEC 60909-0:2016).
satDirectTransX (ShortCircuit)	0..1	PU	Saturated Direct-axis transient reactance. The attribute is primarily used for short circuit calculations according to ANSI.
shortCircuitRotorType (ShortCircuit)	0..1	ShortCircuitRotorKind	Type of rotor, used by short circuit applications, only for single fed short circuit according to IEC 60909.
type	1..1	SynchronousMachineKind	Modes that this synchronous machine can operate in.
voltageRegulationRange (ShortCircuit)	0..1	PerCent	Range of generator voltage regulation (PG in the IEC 60909-0) used for calculation of the impedance correction factor KG defined in IEC 60909-0 This attribute is used to describe the operating voltage of the generating unit.
r (ShortCircuit)	1..1	Resistance	Equivalent resistance (RG) of generator. RG is considered for the calculation of all currents, except for the calculation of the peak current ip. Used for short circuit data exchange according to IEC 60909
x0 (ShortCircuit)	1..1	PU	Zero sequence reactance of the synchronous machine.
x2 (ShortCircuit)	1..1	PU	Negative sequence reactance.
ratedPowerFactor	0..1	Simple_Float	see RotatingMachine
ratedS	0..1	ApparentPower	see RotatingMachine
ratedU	0..1	Voltage	see RotatingMachine
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 279 shows all association ends of SynchronousMachine with other classes.

Table 279 – Association ends of Wires::SynchronousMachine with other classes

mult.	mult.	Class type name	Description
InitialReactiveCapabilityCurve	[0..1]	ReactiveCapabilityCurve	Synchronous machines using this curve as default. Association Based On: Wires:: ReactiveCapabilityCurve.InitialReactiveCapabilityCurve[0..1] ----- Wires:: SynchronousMachine.InitiallyUsedBySynchronousMachines[1..*]
GeneratingUnit	0..1	GeneratingUnit	Inherited Association Based On: Production:: GeneratingUnit.GeneratingUnit[0..1] ----- Wires:: RotatingMachine.RotatingMachine[1..*]
RegulatingControl	0..1	RegulatingControl	Inherited Association Based On: Wires:: RegulatingCondEq.RegulatingCondEq[0..*] ----- Wires:: RegulatingControl.RegulatingControl[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.9.48 TapChanger(Abstract)

Mechanism for changing transformer winding tap positions.

Inheritance path: ->PowerSystemResource->IdentifiedObject

Table 280 shows all attributes of TapChanger.

Table 280 – Attributes of Wires::TapChanger

Attribute name	mult.	Attribute type	Description
highStep	1..1	Integer	Highest possible tap step position, advance from neutral. The attribute shall be greater than lowStep.
lowStep	1..1	Integer	Lowest possible tap step position, retard from neutral
ltcFlag	1..1	Boolean	Specifies whether or not a TapChanger has load tap changing capabilities.
neutralStep	1..1	Integer	The neutral tap step position for this winding. The attribute shall be equal or greater than lowStep and equal or less than highStep.
neutralU	1..1	Voltage	Voltage at which the winding operates at the neutral tap setting.
normalStep	1..1	Integer	The tap step position used in "normal" network operation for this winding. For a "Fixed" tap changer indicates the current physical tap setting. The attribute shall be equal or greater than lowStep and equal or less than highStep.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 281 shows all association ends of TapChanger with other classes.

Table 281 – Association ends of Wires::TapChanger with other classes

mult.	mult.	Class type name	Description
TapChangerControl	[0..1]	TapChangerControl	The tap changers that participates in this regulating tap control scheme. Association Based On: Wires:: TapChanger.TapChanger[0..*] ----- Wires:: TapChangerControl.TapChangerControl[0..1]

6.9.49 TapChangerControl

Describes behavior specific to tap changers, e.g. how the voltage at the end of a line varies with the load level and compensation of the voltage drop by tap adjustment.

Inheritance path: ->RegulatingControl->PowerSystemResource->IdentifiedObject

Table 282 shows all attributes of TapChangerControl.

Table 282 – Attributes of Wires::TapChangerControl

Attribute name	mult.	Attribute type	Description
mode	1..1	RegulatingControlModeKind	see RegulatingControl
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 283 shows all association ends of RegulatingControl with other classes.

Table 283 – Association ends of Wires::RegulatingControl with other classes

mult.	mult.	Class type name	Description
Terminal	1..1	Terminal	Inherited Association Based On: Core:: Terminal.Terminal[0..1] --- Wires:: RegulatingControl.RegulatingControl[0..*]

6.9.50 TapChangerTablePoint

Table 284 shows all attributes of TapChangerTablePoint.

Table 284 – Attributes of Wires::TapChangerTablePoint

Attribute name	mult.	Attribute type	Description
b	0..1	PerCent	The magnetizing branch susceptance deviation in percent of nominal value. The actual susceptance is calculated as follows: calculated magnetizing susceptance = b(nominal) * (1 + b(from this class)/100). The b(nominal) is defined as the static magnetizing susceptance on the associated power transformer end or ends. This model assumes the star impedance (pi model) form.
g	0..1	PerCent	The magnetizing branch conductance deviation in percent of nominal value. The actual conductance is calculated as follows: calculated magnetizing conductance = g(nominal) * (1 + g(from this class)/100). The g(nominal) is defined as the static magnetizing conductance on the associated power transformer end or ends. This model assumes the star impedance (pi model) form.
r	0..1	PerCent	The resistance deviation in percent of nominal value. The actual reactance is calculated as follows: calculated resistance = r(nominal) * (1 + r(from this class)/100). The r(nominal) is defined as the static resistance on the associated power transformer end or ends. This model assumes the star impedance (pi model) form.
ratio	0..1	Simple_Float	The voltage ratio in per unit. Hence this is a value close to one.
step	1..1	Integer	The tap step.
x	0..1	PerCent	The series reactance deviation in percent of nominal value. The actual reactance is calculated as follows: calculated reactance = x(nominal) * (1 + x(from this class)/100). The x(nominal) is defined as the static series reactance on the associated power transformer end or ends. This model assumes the star impedance (pi model) form.

6.9.51 (Operation) TapSchedule

A pre-established pattern over time for a tap step.

Inheritance path: ->SeasonDayTypeSchedule->RegularIntervalSchedule->BasicIntervalSchedule->IdentifiedObject

Table 285 shows all attributes of TapSchedule.

Table 285 – Attributes of Wires::TapSchedule

Attribute name	mult.	Attribute type	Description
timeStep	1..1	Seconds	see RegularIntervalSchedule
endTime	1..1	DateTime	see RegularIntervalSchedule
startTime	1..1	DateTime	see BasicIntervalSchedule
value1Unit	1..1	UnitSymbol	see BasicIntervalSchedule
value2Unit	0..1	UnitSymbol	see BasicIntervalSchedule
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 286 shows all association ends of TapSchedule with other classes.

Table 286 – Association ends of Wires::TapSchedule with other classes

mult.	mult.	Class type name	Description
TapChanger (Operation)	[1..1]	TapChanger	A TapChanger can have TapSchedules. Association Based On: Wires:: TapSchedule.TapSchedules[0..*] ----- Wires:: TapChanger.TapChanger[1]
Season (Operation)	1..1	Season	Inherited Association Based On: LoadModel:: Season.Season[0..1] ----- LoadModel:: SeasonDayTypeSchedule.SeasonDayTypeSchedules[0..*]
DayType (Operation)	1..1	DayType	Inherited Association Based On: LoadModel:: DayType.DayType[0..1] ----- LoadModel:: SeasonDayTypeSchedule.SeasonDayTypeSchedules[0..*]

6.9.52 TransformerEnd(Abstract)

A conducting connection point of a power transformer. It corresponds to a physical transformer winding terminal. In earlier CIM versions, the TransformerWinding class served a similar purpose, but this class is more flexible because it associates to terminal but is not a specialization of ConductingEquipment.

If TransformerEnd.grounded is true, then TransformerEnd.rground and TransformerEnd.xground are required.

Inheritance path: ->IdentifiedObject

Table 287 shows all attributes of TransformerEnd.

Table 287 – Attributes of Wires::TransformerEnd

Attribute name	mult.	Attribute type	Description
rground (ShortCircuit)	0..1	Resistance	(for Yn and Zn connections) Resistance part of neutral impedance where 'grounded' is true.
endNumber	1..1	Integer	Number for this transformer end, corresponding to the end's order in the power transformer vector group or phase angle clock number. Highest voltage winding should be 1. Each end within a power transformer should have a unique subsequent end number. Note the transformer end number need not match the terminal sequence number.
grounded (ShortCircuit)	1..1	Boolean	(for Yn and Zn connections) True if the neutral is solidly grounded.
xground (ShortCircuit)	0..1	Reactance	(for Yn and Zn connections) Reactive part of neutral impedance where 'grounded' is true.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 288 shows all association ends of TransformerEnd with other classes.

Table 288 – Association ends of Wires::TransformerEnd with other classes

mult.	mult.	Class type name	Description
Terminal	[1..1]	Terminal	Terminal of the power transformer to which this transformer end belongs. Association Based On: Wires:: TransformerEnd.TransformerEnd[0..*] ----- Core:: Terminal.Terminal[0..1]
BaseVoltage	[1..1]	BaseVoltage	Base voltage of the transformer end. This is essential for PU calculation. Association Based On: Wires:: TransformerEnd.TransformerEnds[0..*] ----- Core:: BaseVoltage.BaseVoltage[0..1]

6.9.53 Enumerations

6.9.53.1 PetersenCoilModeKind

The mode of operation for a Petersen coil.

Literal	Description	Code
fixed	Fixed position.	
manual	Manual positioning.	
automaticPositioning	Automatic positioning.	

6.9.53.2 RegulatingControlModeKind

The kind of regulation model. For example regulating voltage, reactive power, active power, etc.

Literal	Description	Code
voltage	Voltage is specified.	
activePower	Active power is specified.	
reactivePower	Reactive power is specified.	
currentFlow	Current flow is specified.	
admittance	Admittance is specified.	
timeScheduled	Control switches on/off by time of day. The times may change on the weekend, or in different seasons.	
temperature	Control switches on/off based on the local temperature (i.e., a thermostat).	
powerFactor	Power factor is specified.	

6.9.53.3 ShortCircuitRotorKind

Type of rotor, used by short circuit applications.

Literal	Description	Code
salientPole1	Salient pole 1 in IEC 60909	
salientPole2	Salient pole 2 in IEC 60909	
turboSeries1	Turbo Series 1 in IEC 60909	
turboSeries2	Turbo series 2 in IEC 60909	

6.9.53.4 SVCControlMode

Static VAr Compensator control mode.

Literal	Description	Code
reactivePower		
voltage		

6.9.53.5 SynchronousMachineKind

Synchronous machine type.

Literal	Description	Code
generator		
condenser		
generatorOrCondenser		
motor		
generatorOrMotor		
motorOrCondenser		
generatorOrCondenserOrMotor		

6.9.53.6 TransformerControlMode

Control modes for a transformer.

Literal	Description	Code
volt	Voltage control	
reactive	Reactive power flow control	

6.9.53.7 WindingConnection

Winding connection type.

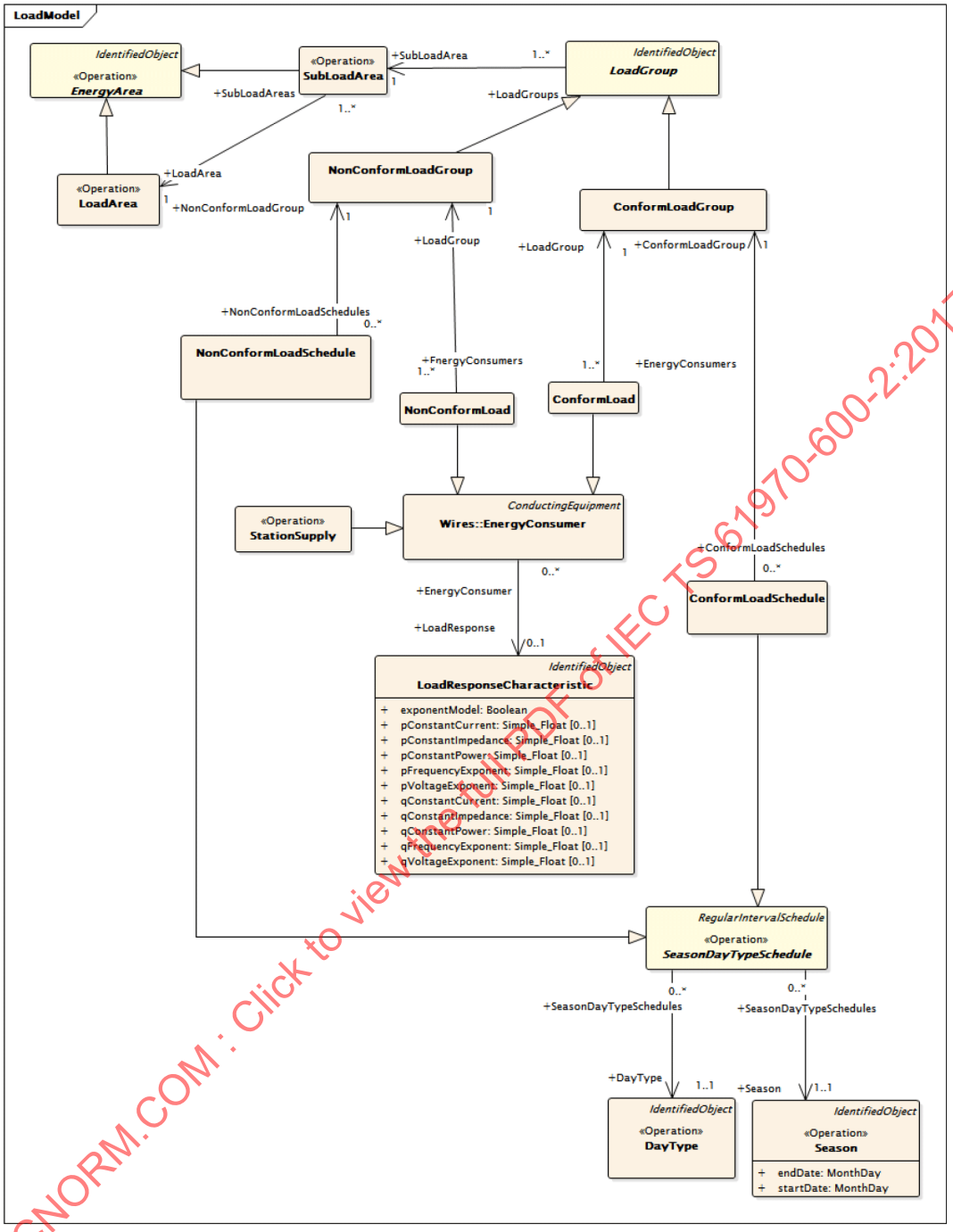
Literal	Description	Code
D	Delta	
Y	Wye	
Z	ZigZag	
Yn	Wye, with neutral brought out for grounding.	
Zn	ZigZag, with neutral brought out for grounding.	
A	Autotransformer common winding	
I	Independent winding, for single-phase connections	

6.10 LoadModel

6.10.1 General

This package is responsible for modeling the energy consumers and the system load as curves and associated curve data. Special circumstances that may affect the load, such as seasons and daytypes, are also included here. This information is used by Load Forecasting and Load Management.

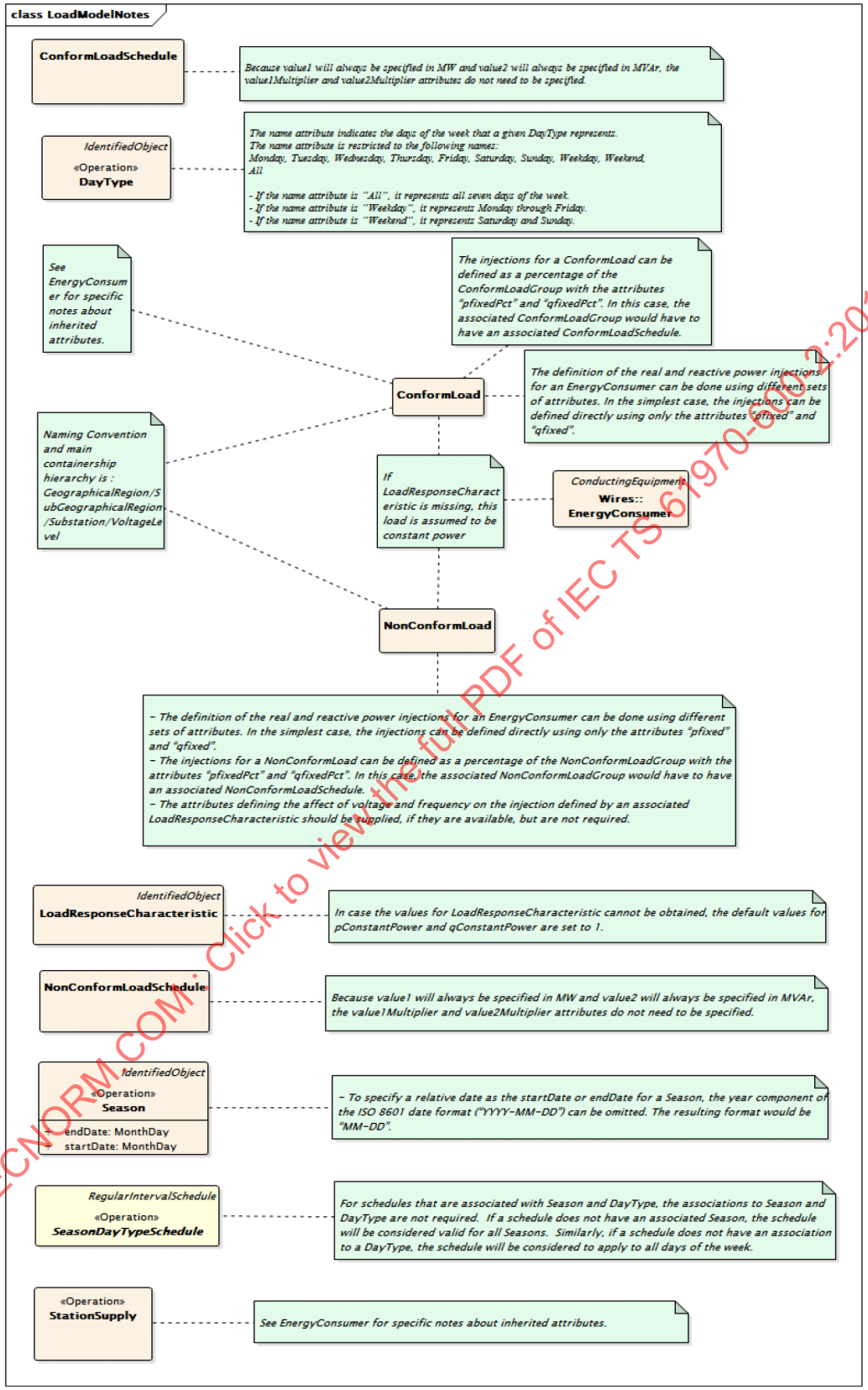
Figure 23 shows the diagram.



IEC

Figure 23 – diagram LoadModel

Figure 24 shows the diagram.



IEC

Figure 24 – diagram LoadModelNotes

6.10.2 ConformLoad

ConformLoad represent loads that follow a daily load change pattern where the pattern can be used to scale the load with a system load.

The injections for a ConformLoad can be defined as a percentage of the ConformLoadGroup with the attributes “pfixedPct” and “qfixedPct”. In this case, the associated ConformLoadGroup would have to have an associated ConformLoadSchedule.

The definition of the real and reactive power injections for an EnergyConsumer can be done using different sets of attributes. In the simplest case, the injections can be defined directly using only the attributes “pfixed” and “qfixed”.

Naming Convention and main containership hierarchy is:
GeographicalRegion/SubGeographicalRegion/Substation/VoltageLevel

See EnergyConsumer for specific notes about inherited attributes.

If LoadResponseCharacteristic is missing, this load is assumed to be constant power.

Inheritance path: ->EnergyConsumer->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 289 shows all attributes of ConformLoad.

Table 289 – Attributes of LoadModel::ConformLoad

Attribute name	mult.	Attribute type	Description
pfixed (Operation)	0..1	ActivePower	see EnergyConsumer
pfixedPct (Operation)	0..1	PerCent	see EnergyConsumer
qfixed (Operation)	0..1	ReactivePower	see EnergyConsumer
qfixedPct (Operation)	0..1	PerCent	see EnergyConsumer
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 290 shows all association ends of ConformLoad with other classes.

Table 290 – Association ends of LoadModel::ConformLoad with other classes

mult.	mult.	Class type name	Description
LoadGroup	[1..1]	ConformLoadGroup	Group of this ConformLoad. Association Based On: LoadModel:: ConformLoadGroup.LoadGroup[0..1] ----- LoadModel:: ConformLoad.EnergyConsumers[0..*]
LoadResponse	0..1	LoadResponseCharacteristic	Inherited Association Based On: Wires:: EnergyConsumer.EnergyConsumer[0..*] ----- LoadModel:: LoadResponseCharacteristic.LoadResponse[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.10.3 ConformLoadGroup

A group of loads conforming to an allocation pattern.

Inheritance path: ->LoadGroup->IdentifiedObject

Table 291 shows all attributes of ConformLoadGroup.

Table 291 – Attributes of LoadModel::ConformLoadGroup

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 292 shows all association ends of LoadGroup with other classes.

Table 292 – Association ends of LoadModel::LoadGroup with other classes

mult.	mult.	Class type name	Description
SubLoadArea	1..1	SubLoadArea	Inherited Association Based On: LoadModel:: SubLoadArea.SubLoadArea[1] ----- LoadModel:: LoadGroup.LoadGroups[1..*]

6.10.4 ConformLoadSchedule

A curve of load versus time (X-axis) showing the active power values (Y1-axis) and reactive power (Y2-axis) for each unit of the period covered. This curve represents a typical pattern of load over the time period for a given day type and season.

Because value1 will always be specified in MW and value2 will always be specified in MVAr, the value1Multiplier and value2Multiplier attributes do not need to be specified.

Inheritance path: ->SeasonDayTypeSchedule->RegularIntervalSchedule->BasicIntervalSchedule->IdentifiedObject

Table 293 shows all attributes of ConformLoadSchedule.

Table 293 – Attributes of LoadModel::ConformLoadSchedule

Attribute name	mult.	Attribute type	Description
timeStep	1..1	Seconds	see RegularIntervalSchedule
endTime	1..1	DateTime	see RegularIntervalSchedule
startTime	1..1	DateTime	see BasicIntervalSchedule
value1Unit	1..1	UnitSymbol	see BasicIntervalSchedule
value2Unit	0..1	UnitSymbol	see BasicIntervalSchedule
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 294 shows all association ends of ConformLoadSchedule with other classes.

Table 294 – Association ends of LoadModel::ConformLoadSchedule with other classes

mult.	mult.	Class type name	Description
ConformLoadGroup	[1..1]	ConformLoadGroup	The ConformLoadGroup where the ConformLoadSchedule belongs. Association Based On: LoadModel:: ConformLoadGroup.ConformLoadGroup[1] ----- LoadModel:: ConformLoadSchedule.ConformLoadSchedules[1..*]
Season (Operation)	1..1	Season	Inherited Association Based On: LoadModel:: Season.Season[0..1] ----- LoadModel:: SeasonDayTypeSchedule.SeasonDayTypeSchedules[0..*]
DayType (Operation)	1..1	DayType	Inherited Association Based On: LoadModel:: DayType.DayType[0..1] ----- LoadModel:: SeasonDayTypeSchedule.SeasonDayTypeSchedules[0..*]

6.10.5 (Operation) DayType

Group of similar days. For example it could be used to represent weekdays, weekend, or holidays.

The name attribute indicates the days of the week that a given DayType represents. The name attribute is restricted to the following names: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, Weekday, Weekend, All – If the name attribute is “All”, it represents all seven days of the week. – If the name attribute is “Weekday”, it represents Monday through Friday. – If the name attribute is “Weekend”, it represents Saturday and Sunday.

Inheritance path: ->IdentifiedObject

Table 295 shows all attributes of DayType.

Table 295 – Attributes of LoadModel::DayType

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.10.6 (Operation) EnergyArea(Abstract)

Describes an area having energy production or consumption. Specializations are intended to support the load allocation function as typically required in energy management systems or planning studies to allocate hypothesized load levels to individual load points for power flow analysis. Often the energy area can be linked to both measured and forecast load levels.

Inheritance path: ->IdentifiedObject

Table 296 shows all attributes of EnergyArea.

Table 296 – Attributes of LoadModel::EnergyArea

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.10.7 (Operation) LoadArea

The class is the root or first level in a hierarchical structure for grouping of loads for the purpose of load flow load scaling.

Inheritance path: ->EnergyArea->IdentifiedObject

Table 297 shows all attributes of LoadArea.

Table 297 – Attributes of LoadModel::LoadArea

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.10.8 LoadGroup(Abstract)

The class is the third level in a hierarchical structure for grouping of loads for the purpose of load flow load scaling.

Inheritance path: ->IdentifiedObject

Table 298 shows all attributes of LoadGroup.

Table 298 – Attributes of LoadModel::LoadGroup

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 299 shows all association ends of LoadGroup with other classes.

Table 299 – Association ends of LoadModel::LoadGroup with other classes

mult.	mult.	Class type name	Description
SubLoadArea	[1..1]	SubLoadArea	The SubLoadArea where the Loadgroup belongs. Association Based On: LoadModel:: SubLoadArea.SubLoadArea[1] ----- LoadModel:: LoadGroup.LoadGroups[1..*]

6.10.9 LoadResponseCharacteristic

Models the characteristic response of the load demand due to changes in system conditions such as voltage and frequency. This is not related to demand response. If LoadResponseCharacteristic.exponentModel is True, the voltage exponents are specified and used as to calculate:

Active power component = Pnominal * (Voltage/cim:BaseVoltage.nominalVoltage) ** cim:LoadResponseCharacteristic.pVoltageExponent
Reactive power component = Qnominal * (Voltage/cim:BaseVoltage.nominalVoltage)**

cim:LoadResponseCharacteristic.qVoltageExponent
Where * means "multiply" and ** is "raised to power of".

OCL constraint: If LoadResponseCharacteristic.exponentModel is false, the portion of reactive power modelled as constant current must be specified

OCL constraint: If LoadResponseCharacteristic.exponentModel is false, the portion of reactive power modelled as constant impedance must be specified

OCL constraint: If LoadResponseCharacteristic.exponentModel is false, the portion of reactive power modelled as constant power must be specified

OCL constraint: If LoadResponseCharacteristic.exponentModel is false, the portion of real power modelled as constant current must be specified

OCL constraint: If LoadResponseCharacteristic.exponentModel is false, the portion of real power modelled as constant impedance must be specified

OCL constraint: If LoadResponseCharacteristic.exponentModel is false, the portion of real power modelled as constant power must be specified

OCL constraint: If LoadResponseCharacteristic.exponentModel is true the Exponent of per unit voltage effecting reactive power must be specified

OCL constraint: If LoadResponseCharacteristic.exponentModel is true the Exponent of per unit voltage effecting real power must be specified

In case the values for LoadResponseCharacteristic cannot be obtained, the default values for pConstantPower and qConstantPower are set to 1.

Inheritance path: ->IdentifiedObject

Table 300 shows all attributes of LoadResponseCharacteristic.

Table 300 – Attributes of LoadModel::LoadResponseCharacteristic

Attribute name	mult.	Attribute type	Description
exponentModel	1..1	Boolean	Indicates the exponential voltage dependency model is to be used. If false, the coefficient model is to be used. The exponential voltage dependency model consist of the attributes - pVoltageExponent - qVoltageExponent. The coefficient model consist of the attributes - pConstantImpedance - pConstantCurrent - pConstantPower - qConstantImpedance - qConstantCurrent - qConstantPower. The sum of pConstantImpedance, pConstantCurrent and pConstantPower shall equal 1. The sum of qConstantImpedance, qConstantCurrent and qConstantPower shall equal 1.
pConstantCurrent	0..1	Simple_Float	Portion of active power load modeled as constant current.
pConstantImpedance	0..1	Simple_Float	Portion of active power load modeled as constant impedance.
pConstantPower	0..1	Simple_Float	Portion of active power load modeled as constant power.
pFrequencyExponent	0..1	Simple_Float	Exponent of per unit frequency effecting active power.
pVoltageExponent	0..1	Simple_Float	Exponent of per unit voltage effecting real power.
qConstantCurrent	0..1	Simple_Float	Portion of reactive power load modeled as constant current.
qConstantImpedance	0..1	Simple_Float	Portion of reactive power load modeled as constant impedance.
qConstantPower	0..1	Simple_Float	Portion of reactive power load modeled as constant power.
qFrequencyExponent	0..1	Simple_Float	Exponent of per unit frequency effecting reactive power.
qVoltageExponent	0..1	Simple_Float	Exponent of per unit voltage effecting reactive power.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.10.10 NonConformLoad

NonConformLoad represent loads that do not follow a daily load change pattern and changes are not correlated with the daily load change pattern.

- The definition of the real and reactive power injections for an EnergyConsumer can be done using different sets of attributes. In the simplest case, the injections can be defined directly using only the attributes “pfixed” and “qfixed”.
- The injections for a NonConformLoad can be defined as a percentage of the NonConformLoadGroup with the attributes “pfixedPct” and “qfixedPct”. In this case, the associated NonConformLoadGroup would have to have an associated NonConformLoadSchedule.

- The attributes defining the affect of voltage and frequency on the injection defined by an associated LoadResponseCharacteristic should be supplied, if they are available, but are not required.

Naming Convention and main containership hierarchy is: GeographicalRegion/SubGeographicalRegion/Substation/VoltageLevel

If LoadResponseCharacteristic is missing, this load is assumed to be constant power

Inheritance path: ->EnergyConsumer->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 301 shows all attributes of NonConformLoad.

Table 301 – Attributes of LoadModel::NonConformLoad

Attribute name	mult.	Attribute type	Description
pfixed (Operation)	0..1	ActivePower	see EnergyConsumer
pfixedPct (Operation)	0..1	PerCent	see EnergyConsumer
qfixed (Operation)	0..1	ReactivePower	see EnergyConsumer
qfixedPct (Operation)	0..1	PerCent	see EnergyConsumer
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 302 shows all association ends of NonConformLoad with other classes.

Table 302 – Association ends of LoadModel::NonConformLoad with other classes

mult.	mult.	Class type name	Description
LoadGroup	[1..1]	NonConformLoadGroup	Conform loads assigned to this ConformLoadGroup. Association Based On: LoadModel:: NonConformLoadGroup.LoadGroup[0..1] ----- LoadModel:: NonConformLoad.EnergyConsumers[0..*]
LoadResponse	0..1	LoadResponseCharacteristic	Inherited Association Based On: Wires:: EnergyConsumer.EnergyConsumer[0..*] ----- LoadModel:: LoadResponseCharacteristic.LoadResponse[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.10.11 NonConformLoadGroup

Loads that do not follow a daily and seasonal load variation pattern.

Inheritance path: ->LoadGroup->IdentifiedObject

Table 303 shows all attributes of NonConformLoadGroup.

Table 303 – Attributes of LoadModel::NonConformLoadGroup

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 304 shows all association ends of LoadGroup with other classes.

Table 304 – Association ends of LoadModel::LoadGroup with other classes

mult.	mult.	Class type name	Description
SubLoadArea	1..1	SubLoadArea	Inherited Association Based On: LoadModel:: SubLoadArea.SubLoadArea[1] --- LoadModel:: LoadGroup.LoadGroups[1..*]

6.10.12 NonConformLoadSchedule

An active power (Y1-axis) and reactive power (Y2-axis) schedule (curves) versus time (X-axis) for non-conforming loads, e.g., large industrial load or power station service (where modeled).

Because value1 will always be specified in MW and value2 will always be specified in MVAr, the value1Multiplier and value2Multiplier attributes do not need to be specified.

Inheritance path: ->SeasonDayTypeSchedule->RegularIntervalSchedule->BasicIntervalSchedule->IdentifiedObject

Table 305 shows all attributes of NonConformLoadSchedule.

Table 305 – Attributes of LoadModel::NonConformLoadSchedule

Attribute name	mult.	Attribute type	Description
timeStep	1..1	Seconds	see RegularIntervalSchedule
endTime	1..1	DateTime	see RegularIntervalSchedule
startTime	1..1	DateTime	see BasicIntervalSchedule
value1Unit	1..1	UnitSymbol	see BasicIntervalSchedule
value2Unit	0..1	UnitSymbol	see BasicIntervalSchedule
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 306 shows all association ends of NonConformLoadSchedule with other classes.

Table 306 – Association ends of LoadModel::NonConformLoadSchedule with other classes

mult.	mult.	Class type name	Description
NonConformLoadGroup	[1..1]	NonConformLoadGroup	The NonConformLoadGroup where the NonConformLoadSchedule belongs. Association Based On: LoadModel:: NonConformLoadGroup.NonConformLoadGroup[1] ----- LoadModel:: NonConformLoadSchedule.NonConformLoadSchedules[1..*]
Season (Operation)	1..1	Season	Inherited Association Based On: LoadModel:: Season.Season[0..1] ----- LoadModel:: SeasonDayTypeSchedule.SeasonDayTypeSchedules[0..*]
DayType (Operation)	1..1	DayType	Inherited Association Based On: LoadModel:: DayType.DayType[0..1] ----- LoadModel:: SeasonDayTypeSchedule.SeasonDayTypeSchedules[0..*]

6.10.13 (Operation) Season

A specified time period of the year.

- To specify a relative date as the startDate or endDate for a Season, the year component of the ISO 8601 date format (“YYYY-MM-DD”) can be omitted. The resulting format would be “MM-DD”.

Inheritance path: ->IdentifiedObject

Table 307 shows all attributes of Season.

Table 307 – Attributes of LoadModel::Season

Attribute name	mult.	Attribute type	Description
endDate (Operation)	1..1	MonthDay	Date season ends.
startDate (Operation)	1..1	MonthDay	Date season starts.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.10.14 (Operation) SeasonDayTypeSchedule(Abstract)

A time schedule covering a 24 hour period, with curve data for a specific type of season and day.

For schedules that are associated with Season and DayType, the associations to Season and DayType are not required. If a schedule does not have an associated Season, the schedule will be considered valid for all Seasons. Similarly, if a schedule does not have an association to a DayType, the schedule will be considered to apply to all days of the week.

Inheritance path: ->RegularIntervalSchedule->BasicIntervalSchedule->IdentifiedObject

Table 308 shows all attributes of SeasonDayTypeSchedule.

Table 308 – Attributes of LoadModel::SeasonDayTypeSchedule

Attribute name	mult.	Attribute type	Description
timeStep	1..1	Seconds	see RegularIntervalSchedule
endTime	1..1	DateTime	see RegularIntervalSchedule
startTime	1..1	DateTime	see BasicIntervalSchedule
value1Unit	1..1	UnitSymbol	see BasicIntervalSchedule
value2Unit	0..1	UnitSymbol	see BasicIntervalSchedule
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 309 shows all association ends of SeasonDayTypeSchedule with other classes.

Table 309 – Association ends of LoadModel::SeasonDayTypeSchedule with other classes

mult.	mult.	Class type name	Description
Season (Operation)	[1..1]	Season	Schedules that use this Season. Association Based On: LoadModel:: Season.Season[0..1] ----- LoadModel:: SeasonDayTypeSchedule.SeasonDayTypeSchedules[0..*]
DayType (Operation)	[1..1]	DayType	Schedules that use this DayType. Association Based On: LoadModel:: DayType.DayType[0..1] ----- LoadModel:: SeasonDayTypeSchedule.SeasonDayTypeSchedules[0..*]

6.10.15 (Operation) StationSupply

Station supply with load derived from the station output.

See EnergyConsumer for specific notes about inherited attributes.

Inheritance path: ->EnergyConsumer->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 310 shows all attributes of StationSupply.

Table 310 – Attributes of LoadModel::StationSupply

Attribute name	mult.	Attribute type	Description
pfixed (Operation)	0..1	ActivePower	see EnergyConsumer
pfixedPct (Operation)	0..1	PerCent	see EnergyConsumer
qfixed (Operation)	0..1	ReactivePower	see EnergyConsumer
qfixedPct (Operation)	0..1	PerCent	see EnergyConsumer
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 311 shows all association ends of EnergyConsumer with other classes.

Table 311 – Association ends of Wires::EnergyConsumer with other classes

mult.	mult.	Class type name	Description
LoadResponse	0..1	LoadResponseCharacteristic	Inherited Association Based On: Wires:: EnergyConsumer.EnergyConsumer[0..*] ----- LoadModel:: LoadResponseCharacteristic.LoadResponse[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.10.16 (Operation) SubLoadArea

The class is the second level in a hierarchical structure for grouping of loads for the purpose of load flow load scaling.

Inheritance path: ->EnergyArea->IdentifiedObject

Table 312 shows all attributes of SubLoadArea.

Table 312 – Attributes of LoadModel::SubLoadArea

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 313 shows all association ends of SubLoadArea with other classes.

Table 313 – Association ends of LoadModel::SubLoadArea with other classes

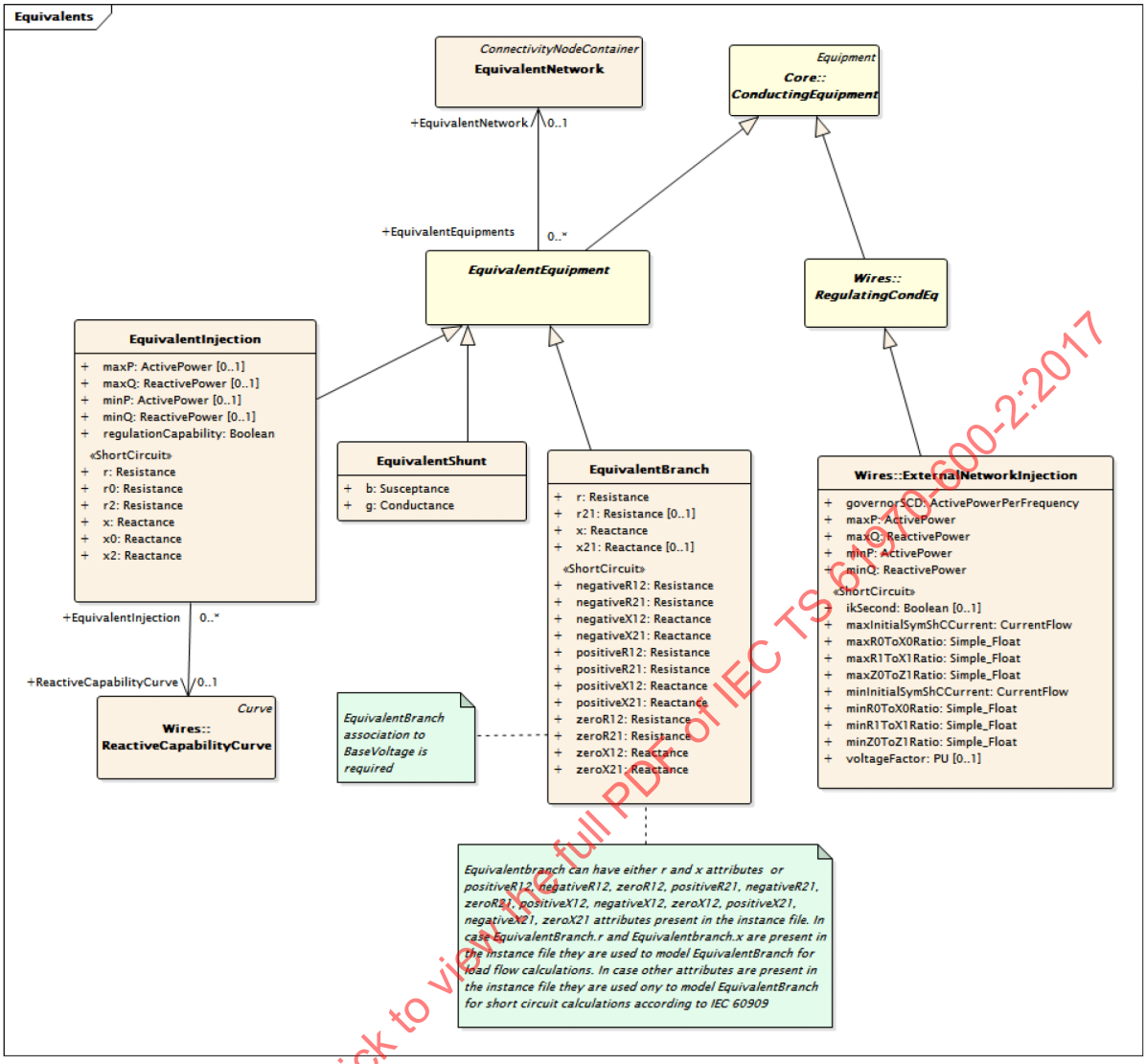
mult.	mult.	Class type name	Description
LoadArea (Operation)	[1..1]	LoadArea	The LoadArea where the SubLoadArea belongs. Association Based On: LoadModel:: LoadArea.LoadArea[1] ----- LoadModel:: SubLoadArea.SubLoadAreas[1..*]

6.11 Equivalents

6.11.1 General

The equivalents package models equivalent networks.

Figure 25 shows the diagram.



IEC

Figure 25 – diagram Equivalents

6.11.2 EquivalentBranch

The class represents equivalent branches.

OCL constraint:If a device has more than one terminal it must have a sequenceNumber

Equivalentbranch can have either r and x attributes or positiveR12, negativeR12, zeroR12, positiveR21, negativeR21, zeroR21, positiveX12, negativeX12, zeroX12, positiveX21, negativeX21, zeroX21 attributes present in the instance file. In case EquivalentBranch.r and Equivalentbranch.x are present in the instance file they are used to model EquivalentBranch for load flow calculations. In case other attributes are present in the instance file they are used ony to model EquivalentBranch for short circuit calculations according to IEC 60909.

EquivalentBranch association to BaseVoltage is required.

Inheritance path: ->EquivalentEquipment->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 314 shows all attributes of EquivalentBranch.

Table 314 – Attributes of Equivalents::EquivalentBranch

Attribute name	mult.	Attribute type	Description
negativeR12 (ShortCircuit)	1..1	Resistance	Negative sequence series resistance from terminal sequence 1 to terminal sequence 2. Used for short circuit data exchange according to IEC 60909 EquivalentBranch is a result of network reduction prior to the data exchange
negativeR21 (ShortCircuit)	1..1	Resistance	Negative sequence series resistance from terminal sequence 2 to terminal sequence 1. Used for short circuit data exchange according to IEC 60909 EquivalentBranch is a result of network reduction prior to the data exchange
negativeX12 (ShortCircuit)	1..1	Reactance	Negative sequence series reactance from terminal sequence 1 to terminal sequence 2. Used for short circuit data exchange according to IEC 60909 Usage: EquivalentBranch is a result of network reduction prior to the data exchange
negativeX21 (ShortCircuit)	1..1	Reactance	Negative sequence series reactance from terminal sequence 2 to terminal sequence 1. Used for short circuit data exchange according to IEC 60909. Usage: EquivalentBranch is a result of network reduction prior to the data exchange
positiveR12 (ShortCircuit)	1..1	Resistance	Positive sequence series resistance from terminal sequence 1 to terminal sequence 2 . Used for short circuit data exchange according to IEC 60909. EquivalentBranch is a result of network reduction prior to the data exchange.
positiveR21 (ShortCircuit)	1..1	Resistance	Positive sequence series resistance from terminal sequence 2 to terminal sequence 1. Used for short circuit data exchange according to IEC 60909 EquivalentBranch is a result of network reduction prior to the data exchange
positiveX12 (ShortCircuit)	1..1	Reactance	Positive sequence series reactance from terminal sequence 1 to terminal sequence 2. Used for short circuit data exchange according to IEC 60909 Usage: EquivalentBranch is a result of network reduction prior to the data exchange
positiveX21 (ShortCircuit)	1..1	Reactance	Positive sequence series reactance from terminal sequence 2 to terminal sequence 1. Used for short circuit data exchange according to IEC 60909 Usage: EquivalentBranch is a result of network reduction prior to the data exchange
r	1..1	Resistance	Positive sequence series resistance of the reduced branch.
r21	0..1	Resistance	Resistance from terminal sequence 2 to terminal sequence 1 .Used for steady state power flow. This attribute is optional and represent unbalanced network such as off-nominal phase shifter. If only EquivalentBranch.r is given, then EquivalentBranch.r21 is assumed equal to EquivalentBranch.r. Usage rule: EquivalentBranch is a result of network reduction prior to the data exchange.
x	1..1	Reactance	Positive sequence series reactance of the reduced branch.
x21	0..1	Reactance	Reactance from terminal sequence 2 to terminal sequence 1 .Used for steady state power flow. This attribute is optional and represent unbalanced network such as off-nominal phase shifter. If only EquivalentBranch.x is given, then EquivalentBranch.x21 is assumed equal to EquivalentBranch.x. Usage rule: EquivalentBranch is a result of network reduction prior to the data exchange.

Attribute name	mult.	Attribute type	Description
zeroR12 (ShortCircuit)	1..1	Resistance	Zero sequence series resistance from terminal sequence 1 to terminal sequence 2. Used for short circuit data exchange according to IEC 60909 EquivalentBranch is a result of network reduction prior to the data exchange
zeroR21 (ShortCircuit)	1..1	Resistance	Zero sequence series resistance from terminal sequence 2 to terminal sequence 1. Used for short circuit data exchange according to IEC 60909 Usage: EquivalentBranch is a result of network reduction prior to the data exchange
zeroX12 (ShortCircuit)	1..1	Reactance	Zero sequence series reactance from terminal sequence 1 to terminal sequence 2. Used for short circuit data exchange according to IEC 60909 Usage: EquivalentBranch is a result of network reduction prior to the data exchange
zeroX21 (ShortCircuit)	1..1	Reactance	Zero sequence series reactance from terminal sequence 2 to terminal sequence 1. Used for short circuit data exchange according to IEC 60909 Usage: EquivalentBranch is a result of network reduction prior to the data exchange
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 315 shows all association ends of EquivalentEquipment with other classes.

Table 315 – Association ends of Equivalents::EquivalentEquipment with other classes

mult.	mult.	Class type name	Description
EquivalentNetwork	0..1	EquivalentNetwork	Inherited Association Based On: Equivalents:: EquivalentEquipment.EquivalentEquipments[0..*] ----- Equivalents:: EquivalentNetwork.EquivalentNetwork[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.11.3 EquivalentEquipment(Abstract)

The class represents equivalent objects that are the result of a network reduction. The class is the base for equivalent objects of different types.

Inheritance path: ->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 316 shows all attributes of EquivalentEquipment.

Table 316 – Attributes of Equivalents::EquivalentEquipment

Attribute name	mult.	Attribute type	Description
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 317 shows all association ends of EquivalentEquipment with other classes.

Table 317 – Association ends of Equivalents::EquivalentEquipment with other classes

mult.	mult.	Class type name	Description
EquivalentNetwork	[0..1]	EquivalentNetwork	The associated reduced equivalents. Association Based On: Equivalents:: EquivalentEquipment.EquivalentEquipments[0..*] ----- Equivalents:: EquivalentNetwork.EquivalentNetwork[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.11.4 EquivalentInjection

This class represents equivalent injections (generation or load). Voltage regulation is allowed only at the point of connection.

Inheritance path: ->EquivalentEquipment->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 318 shows all attributes of EquivalentInjection.

Table 318 – Attributes of Equivalents::EquivalentInjection

Attribute name	mult.	Attribute type	Description
maxP	0..1	ActivePower	Maximum active power of the injection.
maxQ	0..1	ReactivePower	Used for modeling of infeed for load flow exchange. Not used for short circuit modeling. If maxQ and minQ are not used ReactiveCapabilityCurve can be used.
minP	0..1	ActivePower	Minimum active power of the injection.
minQ	0..1	ReactivePower	Used for modeling of infeed for load flow exchange. Not used for short circuit modeling. If maxQ and minQ are not used ReactiveCapabilityCurve can be used.
r (ShortCircuit)	1..1	Resistance	Positive sequence resistance. Used to represent Extended-Ward (IEC 60909). Usage: Extended-Ward is a result of network reduction prior to the data exchange.
r0 (ShortCircuit)	1..1	Resistance	Zero sequence resistance. Used to represent Extended-Ward (IEC 60909). Usage: Extended-Ward is a result of network reduction prior to the data exchange.
r2 (ShortCircuit)	1..1	Resistance	Negative sequence resistance. Used to represent Extended-Ward (IEC 60909). Usage: Extended-Ward is a result of network reduction prior to the data exchange.
regulationCapability	1..1	Boolean	Specifies whether or not the EquivalentInjection has the capability to regulate the local voltage.
x (ShortCircuit)	1..1	Reactance	Positive sequence reactance. Used to represent Extended-Ward (IEC 60909). Usage: Extended-Ward is a result of network reduction prior to the data exchange.
x0 (ShortCircuit)	1..1	Reactance	Zero sequence reactance. Used to represent Extended-Ward (IEC 60909). Usage: Extended-Ward is a result of network reduction prior to the data exchange.
x2 (ShortCircuit)	1..1	Reactance	Negative sequence reactance. Used to represent Extended-Ward (IEC 60909). Usage: Extended-Ward is a result of network reduction prior to the data exchange.
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 319 shows all association ends of EquivalentInjection with other classes.

Table 319 – Association ends of Equivalents::EquivalentInjection with other classes

mult.	mult.	Class type name	Description
ReactiveCapabilityCurve	[0..1]	ReactiveCapabilityCurve	The equivalent injection using this reactive capability curve. Association Based On: Wires:: ReactiveCapabilityCurve.ReactiveCapabilityCurve[0..1] ----- Equivalents:: EquivalentInjection.EquivalentInjection[0..*]
EquivalentNetwork	0..1	EquivalentNetwork	Inherited Association Based On: Equivalents:: EquivalentEquipment.EquivalentEquipments[0..*] ----- Equivalents:: EquivalentNetwork.EquivalentNetwork[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.11.5 EquivalentNetwork

A class that represents an external meshed network that has been reduced to an electrically equivalent model. The ConnectivityNodes contained in the equivalent are intended to reflect internal nodes of the equivalent. The boundary Connectivity nodes where the equivalent connects outside itself are NOT contained by the equivalent.

Inheritance path: ->ConnectivityNodeContainer->PowerSystemResource->IdentifiedObject

Table 320 shows all attributes of EquivalentNetwork.

Table 320 – Attributes of Equivalents::EquivalentNetwork

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

6.11.6 EquivalentShunt

The class represents equivalent shunts.

Inheritance path: ->EquivalentEquipment->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 321 shows all attributes of EquivalentShunt.

Table 321 – Attributes of Equivalents::EquivalentShunt

Attribute name	mult.	Attribute type	Description
b	1..1	Susceptance	Positive sequence shunt susceptance.
g	1..1	Conductance	Positive sequence shunt conductance.
aggregate	0..1	Boolean	see Equipment
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 322 shows all association ends of EquivalentEquipment with other classes.

Table 322 – Association ends of Equivalents::EquivalentEquipment with other classes

mult.	mult.	Class type name	Description
EquivalentNetwork	0..1	EquivalentNetwork	Inherited Association Based On: Equivalents:: EquivalentEquipment.EquivalentEquipments[0..*] ----- Equivalents:: EquivalentNetwork.EquivalentNetwork[0..1]
BaseVoltage	0..1	BaseVoltage	Inherited Association Based On: Core:: BaseVoltage.BaseVoltage[0..1] ----- Core:: ConductingEquipment.ConductingEquipment[0..*]
EquipmentContainer	0..1	EquipmentContainer	Inherited Association Based On: Core:: EquipmentContainer.EquipmentContainer[0..1] ----- Core:: Equipment.Equipments[0..*]

6.12 ControlArea

6.12.1 General

The ControlArea package models area specifications which can be used for a variety of purposes. The package as a whole models potentially overlapping control area specifications for the purpose of actual generation control, load forecast area load capture, or powerflow based analysis.

Figure 26 shows the diagram.

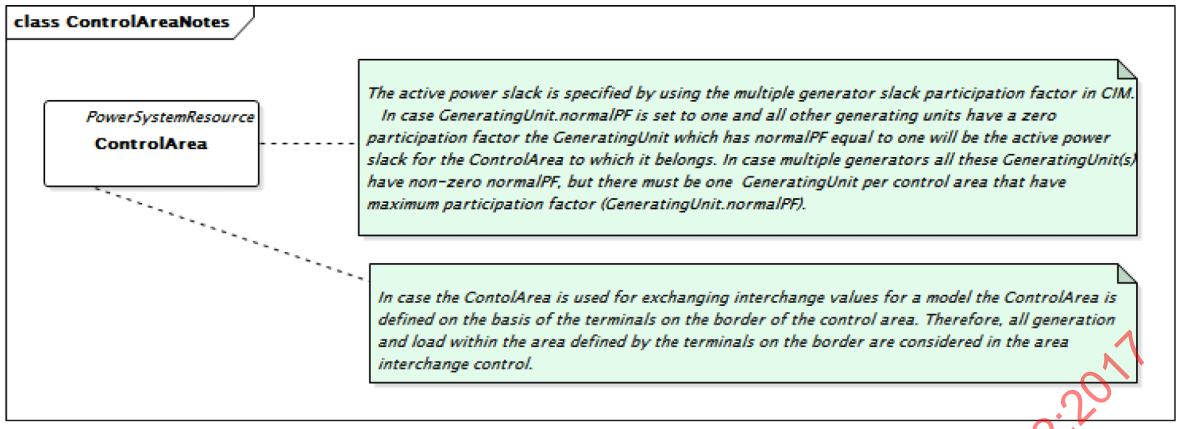


Figure 26 – diagram ControlAreaNotes

Figure 27 shows the diagram.

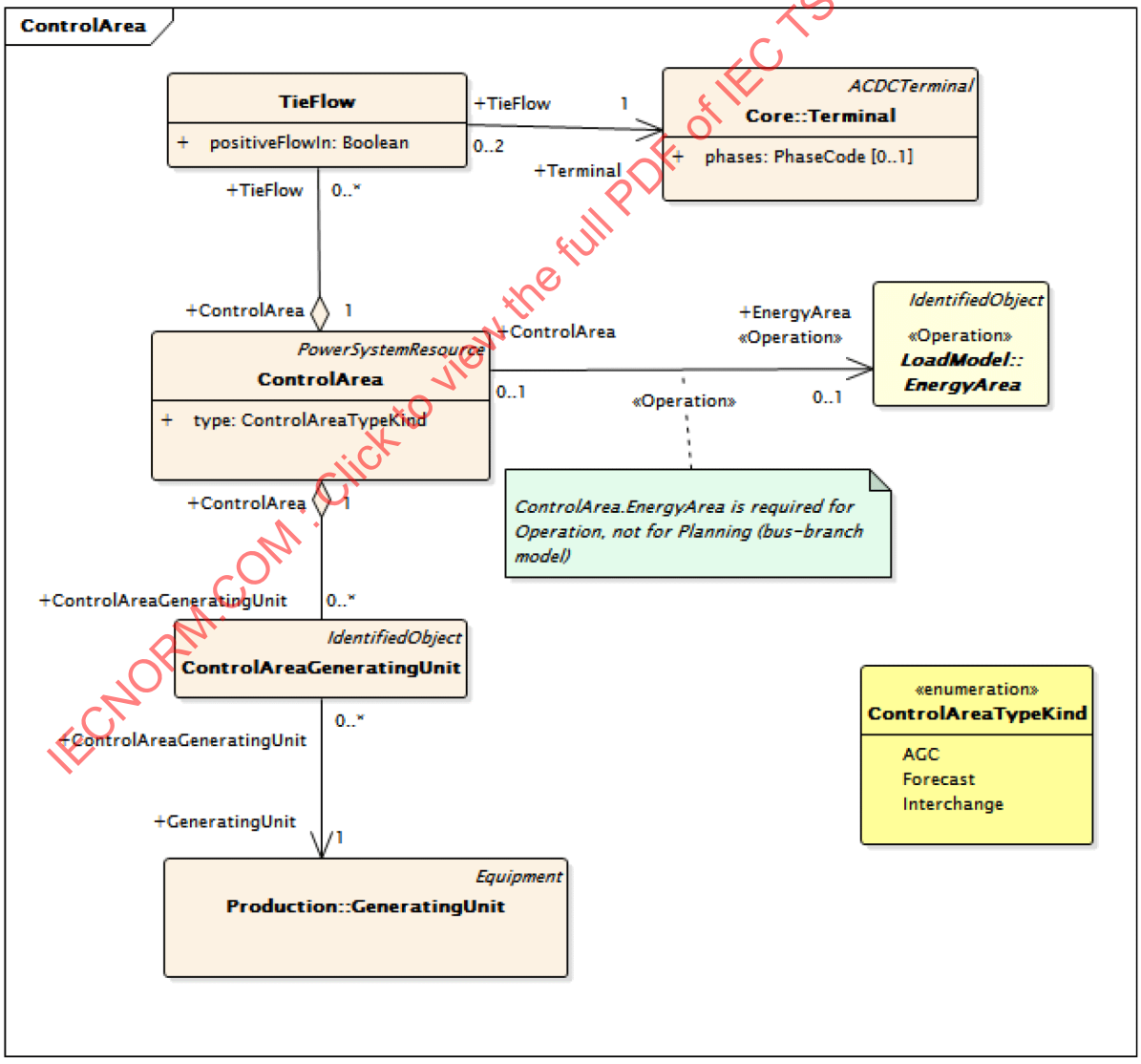


Figure 27 – diagram ControlArea

6.12.2 ControlArea

A control area is a grouping of generating units and/or loads and a cutset of tie lines (as terminals) which may be used for a variety of purposes including automatic generation control, powerflow solution area interchange control specification, and input to load forecasting. Note that any number of overlapping control area specifications can be superimposed on the physical model.

In case the ContolArea is used for exchanging interchange values for a model the ControlArea is defined on the basis of the terminals on the border of the control area. Therefore, all generation and load within the area defined by the terminals on the border are considered in the area interchange control.

The active power slack is specified by using the multiple generator slack participation factor in CIM. In case GeneratingUnit.normalPF is set to one and all other generating units have a zero participation factor the GeneratingUnit which has normalPF equal to one will be the active power slack for the ControlArea to which it belongs. In case multiple generators all these GeneratingUnit(s) have non-zero normalPF, but there must be one GeneratingUnit per control area that have maximum participation factor (GeneratingUnit.normalPF).

Inheritance path: ->PowerSystemResource->IdentifiedObject

Table 323 shows all attributes of ControlArea.

Table 323 – Attributes of ControlArea::ControlArea

Attribute name	mult.	Attribute type	Description
type	1..1	ControlAreaTypeKind	The primary type of control area definition used to determine if this is used for automatic generation control, for planning interchange control, or other purposes. A control area specified with primary type of automatic generation control could still be forecast and used as an interchange area in power flow analysis.
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 324 shows all association ends of ControlArea with other classes.

Table 324 – Association ends of ControlArea::ControlArea with other classes

mult.	mult.	Class type name	Description
EnergyArea (Operation)	[0..1]	EnergyArea	The energy area that is forecast from this control area specification. Association Based On: ControlArea:: ControlArea.ControlArea[0..1] ----- LoadModel:: EnergyArea.EnergyArea[0..1]

6.12.3 ControlAreaGeneratingUnit

A control area generating unit. This class is needed so that alternate control area definitions may include the same generating unit. Note only one instance within a control area should reference a specific generating unit.

Inheritance path: ->IdentifiedObject

Table 325 shows all attributes of ControlAreaGeneratingUnit.

Table 325 – Attributes of ControlArea::ControlAreaGeneratingUnit

Attribute name	mult.	Attribute type	Description
description	0..1	String	see IdentifiedObject
energyIdentCodeEic (Entsoe)	0..1	String	see IdentifiedObject
mRID	0..1	String	see IdentifiedObject
name	1..1	String	see IdentifiedObject
shortName (Entsoe)	0..1	String	see IdentifiedObject

Table 326 shows all association ends of ControlAreaGeneratingUnit with other classes.

Table 326 – Association ends of ControlArea::ControlAreaGeneratingUnit with other classes

mult.	mult.	Class type name	Description
ControlArea	[1..1]	ControlArea	The parent control area for the generating unit specifications. Association Based On: ControlArea:: ControlAreaGeneratingUnit.ControlAreaGeneratingUnit[0..*] ----- ControlArea:: ControlArea.ControlArea[1]
GeneratingUnit	[1..1]	GeneratingUnit	The generating unit specified for this control area. Note that a control area should include a GeneratingUnit only once. Association Based On: ControlArea:: ControlAreaGeneratingUnit.ControlAreaGeneratingUnit[0..*] ----- Production:: GeneratingUnit.GeneratingUnit[1]

6.12.4 TieFlow

A flow specification in terms of location and direction for a control area.

Table 327 shows all attributes of TieFlow.

Table 327 – Attributes of ControlArea::TieFlow

Attribute name	mult.	Attribute type	Description
positiveFlowIn	1..1	Boolean	True if the flow into the terminal (load convention) is also flow into the control area. For example, this attribute should be true if using the tie line terminal further away from the control area. For example to represent a tie to a shunt component (like a load or generator) in another area, this is the near end of a branch and this attribute would be specified as false.

Table 328 shows all association ends of TieFlow with other classes.

Table 328 – Association ends of ControlArea::TieFlow with other classes

mult.	mult.	Class type name	Description
ControlArea	[1..1]	ControlArea	The control area of the tie flows. Association Based On: ControlArea:: TieFlow.TieFlow[0..*] ----- ControlArea:: ControlArea.ControlArea[1]
Terminal	[1..1]	Terminal	The terminal to which this tie flow belongs. Association Based On: ControlArea:: TieFlow.TieFlow[0..2] ----- Core:: Terminal.Terminal[1]

6.12.5 Enumerations – ControlAreaTypeKind

The type of control area.

Literal	Description	Code
AGC	Used for automatic generation control.	
Forecast	Used for load forecast.	
Interchange	Used for interchange specification or control.	

6.13 Global Used Datatypes

6.13.1 ApparentPower

Product of the RMS value of the voltage and the RMS value of the current.

Value type is IEEE 754 simple precision floating point

Table 329 shows all attributes of ApparentPower.

Table 329 – Attributes of DomainProfile::ApparentPower.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= VA
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= M

6.13.2 ActivePower

Product of RMS value of the voltage and the RMS value of the in-phase component of the current.

Value type is IEEE 754 simple precision floating point

Table 330 shows all attributes of ActivePower.

Table 330 – Attributes of DomainProfile::ActivePower.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= W
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= M

6.13.3 Voltage

Electrical voltage, can be both AC and DC.

Value type is IEEE 754 simple precision floating point

Table 331 shows all attributes of Voltage.

Table 331 – Attributes of DomainProfile::Voltage.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= V
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= k

6.13.4 Resistance

Resistance (real part of impedance).

Value type is IEEE 754 simple precision floating point

Table 332 shows all attributes of Resistance.

Table 332 – Attributes of DomainProfile::Resistance.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= ohm
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none

6.13.5 ActivePowerPerCurrentFlow

Table 333 shows all attributes of ActivePowerPerCurrentFlow.

Table 333 – Attributes of DomainProfile::ActivePowerPerCurrentFlow.

Attribute name	mult.	Attribute type	Description
denominatorMultiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none
denominatorUnit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= A
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= M
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= W
value (CIMDatatype)	0..1	Float	

6.13.6 AngleDegrees

Measurement of angle in degrees.

Value type is IEEE 754 simple precision floating point

Table 334 shows all attributes of AngleDegrees.

Table 334 – Attributes of DomainProfile::AngleDegrees.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= deg
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none

6.13.7 CurrentFlow

Electrical current with sign convention: positive flow is out of the conducting equipment into the connectivity node. Can be both AC and DC.

Value type is IEEE 754 simple precision floating point

Table 335 shows all attributes of CurrentFlow.

Table 335 – Attributes of DomainProfile::CurrentFlow.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= A
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none

6.13.8 Inductance

Inductive part of reactance (imaginary part of impedance), at rated frequency.

Table 336 shows all attributes of Inductance.

Table 336 – Attributes of DomainProfile::Inductance.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= H
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none

6.13.9 Capacitance

Capacitive part of reactance (imaginary part of impedance), at rated frequency.

Table 337 shows all attributes of Capacitance.

Table 337 – Attributes of DomainProfile::Capacitance.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= F
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none

6.13.10 Length

Unit of length. Never negative.

Value type is IEEE 754 simple precision floating point

Table 338 shows all attributes of Length.

Table 338 – Attributes of DomainProfile::Length.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= m
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= k

6.13.11 CapacitancePerLength

Capacitance per unit of length.

Table 339 shows all attributes of CapacitancePerLength.

Table 339 – Attributes of DomainProfile::CapacitancePerLength.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= F
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none
denominatorUnit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= m
denominatorMultiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none

6.13.12 InductancePerLength

Inductance per unit of length.

Table 340 shows all attributes of InductancePerLength.

Table 340 – Attributes of DomainProfile::InductancePerLength.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= H
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none
denominatorUnit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= m
denominatorMultiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none

6.13.13 ResistancePerLength

Resistance (real part of impedance) per unit of length.

Table 341 shows all attributes of ResistancePerLength.

Table 341 – Attributes of DomainProfile::ResistancePerLength.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= ohm
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none
denominatorUnit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= m
denominatorMultiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none

6.13.14 Simple_Float

A floating point number. The range is unspecified and not limited.

In an ENTSO-E profile, Simple_Float range is the IEEE 754 simple precision floating point one. It corresponds to xs:float datatype.

Table 342 shows all attributes of Simple_Float.

Table 342 – Attributes of DomainProfile::Simple_Float.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	1..1	Float	

6.13.15 PerCent

Percentage on a defined base. For example, specify as 100 to indicate at the defined base.

Value type is IEEE 754 simple precision floating point

Table 343 shows all attributes of PerCent.

Table 343 – Attributes of DomainProfile::PerCent.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	Normally 0 – 100 on a defined base
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= none
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none

6.13.16 AngleRadians

Phase angle in radians.

Table 344 shows all attributes of AngleRadians.

Table 344 – Attributes of DomainProfile::AngleRadians.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= rad
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none

6.13.17 Reactance

Reactance (imaginary part of impedance), at rated frequency.

Value type is IEEE 754 simple precision floating point

Table 345 shows all attributes of Reactance.

Table 345 – Attributes of DomainProfile::Reactance

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= ohm
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none

6.13.18 Money

Amount of money.

Value atype is IEEE 754 simple precision floating point

Table 346 shows all attributes of Money.

Table 346 – Attributes of DomainProfile::Money.

Attribute name	mult.	Attribute type	Description
unit (CIMDatatype)	0..1	Currency	ConstantValue= EUR
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none
value (CIMDatatype)	0..1	Decimal	

6.13.19 Seconds

Time, in seconds.

Value type is IEEE 754 simple precision floating point

Table 347 shows all attributes of Seconds.

Table 347 – Attributes of DomainProfile::Seconds.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	Time, in seconds
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= s
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none

6.13.20 Susceptance

Imaginary part of admittance.

Value type is IEEE 754 simple precision floating point

Table 348 shows all attributes of Susceptance.

Table 348 – Attributes of DomainProfile::Susceptance.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= S
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none

6.13.21 Conductance

Factor by which voltage must be multiplied to give corresponding power lost from a circuit. Real part of admittance.

Value type is IEEE 754 simple precision floating point

Table 349 shows all attributes of Conductance.

Table 349 – Attributes of DomainProfile::Conductance.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= S
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none

6.13.22 Temperature

Value of temperature in degrees Celsius.

Value type is IEEE 754 simple precision floating point

Table 350 shows all attributes of Temperature.

Table 350 – Attributes of DomainProfile::Temperature.

Attribute name	mult.	Attribute type	Description
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= degC
value (CIMDatatype)	0..1	Float	

6.13.23 Frequency

Cycles per second.

Value type is IEEE 754 simple precision floating point

Table 351 shows all attributes of Frequency.

Table 351 – Attributes of DomainProfile::Frequency.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= Hz
multiplier (CIMDatatype)	0..1	UnitMultiplier	DefaultValue= none

6.13.24 RotationSpeed

Number of revolutions per second.

Value type is IEEE 754 simple precision floating point

Table 352 shows all attributes of RotationSpeed.

Table 352 – Attributes of DomainProfile::RotationSpeed.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= none
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none
denominatorUnit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= s
denominatorMultiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none

6.13.25 ReactivePower

Product of RMS value of the voltage and the RMS value of the quadrature component of the current.

Value type is IEEE 754 simple precision floating point

Table 353 shows all attributes of ReactivePower.

Table 353 – Attributes of DomainProfile::ReactivePower.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= VAr
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= M

6.13.26 ActivePowerPerFrequency

Active power variation with frequency.

Table 354 shows all attributes of ActivePowerPerFrequency.

Table 354 – Attributes of DomainProfile::ActivePowerPerFrequency.

Attribute name	mult.	Attribute type	Description
denominatorMultiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none
denominatorUnit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= Hz
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= M
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= W
value (CIMDatatype)	0..1	Float	

6.13.27 PU

Per Unit – a positive or negative value referred to a defined base. Values typically range from -10 to +10.

Value type is IEEE 754 simple precision floating point

Table 355 shows all attributes of PU.

Table 355 – Attributes of DomainProfile::PU.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= none
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= none

6.13.28 VoltagePerReactivePower

Voltage variation with reactive power.

Value type is IEEE 754 simple precision floating point

Table 356 shows all attributes of VoltagePerReactivePower.

Table 356 – Attributes of DomainProfile::VoltagePerReactivePower.

Attribute name	mult.	Attribute type	Description
value (CIMDatatype)	0..1	Float	
unit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= V
denominatorMultiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= M
multiplier (CIMDatatype)	0..1	UnitMultiplier	ConstantValue= k
denominatorUnit (CIMDatatype)	0..1	UnitSymbol	ConstantValue= VAr

6.13.29 Enumerations

6.13.29.1 UnitSymbol

The units defined for usage in the CIM.

Literal	Description	Code
VA	Apparent power in volt ampere.	
W	Active power in watt.	
VAr	Reactive power in volt ampere reactive.	
VAh	Apparent energy in volt ampere hours.	
Wh	Real energy in what hours.	
VArh	Reactive energy in volt ampere reactive hours.	
V	Voltage in volt.	
ohm	Resistance in ohm.	
A	Current in ampere.	
F	Capacitance in farad.	
H	Inductance in henry.	
degC	Relative temperature in degrees Celsius. In the SI unit system the symbol is ºC. Electric charge is measured in coulomb that has the unit symbol C. To distinguish degree Celsius form coulomb the symbol used in the UML is degC. Reason for not using ºC is the special character º is difficult to manage in software.	
s	Time in seconds.	
min	Time in minutes.	
h	Time in hours.	
deg	Plane angle in degrees.	
rad	Plane angle in radians.	
J	Energy in joule.	
N	Force in newton.	
S	Conductance in siemens.	
none	Dimension less quantity, e.g. count, per unit, etc.	
Hz	Frequency in hertz.	
g	Mass in gram.	
Pa	Pressure in pascal (n/m2).	
m	Length in meter.	
m2	Area in square meters.	
m3	Volume in cubic meters.	

6.13.29.2 UnitMultiplier

The unit multipliers defined for the CIM.

Literal	Description	Code
p	Pico 10**-12.	
n	Nano 10**-9.	
micro	Micro 10**-6.	
m	Milli 10**-3.	
c	Centi 10**-2.	
d	Deci 10**-1.	
k	Kilo 10**3.	
M	Mega 10**6.	
G	Giga 10**9.	
T	Tera 10**12.	
none	No multiplier or equivalently multiply by 1.	

6.13.29.3 UnitSymbol

The units defined for usage in the CIM.

Literal	Description	Code
VA	Apparent power in volt ampere.	
W	Active power in watt.	
VAr	Reactive power in volt ampere reactive.	
VAh	Apparent energy in volt ampere hours.	
Wh	Real energy in what hours.	
VArh	Reactive energy in volt ampere reactive hours.	
V	Voltage in volt.	
ohm	Resistance in ohm.	
A	Current in ampere.	
F	Capacitance in farad.	
H	Inductance in henry.	
degC	Relative temperature in degrees Celsius. In the SI unit system the symbol is °C. Electric charge is measured in coulomb that has the unit symbol C. To distinguish degree Celsius form coulomb the symbol used in the UML is degC. Reason for not using °C is the special character ° is difficult to manage in software.	
s	Time in seconds.	
min	Time in minutes.	
h	Time in hours.	
deg	Plane angle in degrees.	
rad	Plane angle in radians.	
J	Energy in joule.	
N	Force in newton.	
S	Conductance in siemens.	
none	Dimension less quantity, e.g. count, per unit, etc.	
Hz	Frequency in hertz.	
g	Mass in gram.	
Pa	Pressure in pascal (n/m2).	
m	Length in meter.	
m2	Area in square meters.	
m3	Volume in cubic meters.	

6.13.29.4 DCPolarityKind

Polarity for DC circuits.

Literal	Description	Code
positive	Positive pole.	
middle	Middle pole, potentially grounded.	
negative	Negative pole.	

6.13.29.5 DCConverterOperatingModeKind

The operating mode of an HVDC bipole.

Literal	Description	Code
bipolar	Bipolar operation.	
monopolarMetallicReturn	Monopolar operation with metallic return	
monopolarGroundReturn	Monopolar operation with ground return	

6.13.29.6 PhaseCode

Enumeration of phase identifiers. Allows designation of phases for both transmission and distribution equipment, circuits and loads. Residential and small commercial loads are often served from single-phase, or split-phase, secondary circuits. The example of s12N, phases 1 and 2 refer to hot wires that are 180 degrees out of phase, while N refers to the neutral wire. Through single-phase transformer connections, these secondary circuits may be served from one or two of the primary phases A, B, and C. For three-phase loads, use the A, B, C phase codes instead of s12N.

Literal	Description	Code
ABCN	Phases A, B, C, and N.	
ABC	Phases A, B, and C.	
ABN	Phases A, B, and neutral.	
ACN	Phases A, C and neutral.	
BCN	Phases B, C, and neutral.	
AB	Phases A and B.	
AC	Phases A and C.	
BC	Phases B and C.	
AN	Phases A and neutral.	
BN	Phases B and neutral.	
CN	Phases C and neutral.	
A	Phase A.	
B	Phase B.	
C	Phase C.	
N	Neutral phase.	
s1N	Secondary phase 1 and neutral.	
s2N	Secondary phase 2 and neutral.	
s12N	Secondary phases 1, 2, and neutral.	
s1	Secondary phase 1.	
s2	Secondary phase 2.	
s12	Secondary phase 1 and 2.	

6.13.29.7 Source

Source gives information related to the origin of a value.

Literal	Description	Code
PROCESS	The value is provided by input from the process I/O or being calculated from some function.	
DEFAULTED	The value contains a default value.	
SUBSTITUTED	The value is provided by input of an operator or by an automatic source.	

6.13.29.8 Validity

Validity for MeasurementValue.

Literal	Description	Code
GOOD	The value is marked good if no abnormal condition of the acquisition function or the information source is detected.	
QUESTIONABLE	The value is marked questionable if a supervision function detects an abnormal behaviour, however the value could still be valid. The client is responsible for determining whether or not values marked "questionable" should be used.	
INVALID	The value is marked invalid when a supervision function recognises abnormal conditions of the acquisition function or the information source (missing or non-operating updating devices). The value is not defined under this condition. The mark invalid is used to indicate to the client that the value may be incorrect and shall not be used.	

6.13.29.9 FuelType

Type of fuel.

Literal	Description	Code
coal	Generic coal, not including lignite type.	
oil	Oil.	
gas	Natural gas.	
lignite	The fuel is lignite coal. Note that this is a special type of coal, so the other enum of coal is reserved for hard coal types or if the exact type of coal is not known.	
hardCoal	Hard coal.	
oilShale	Oil Shale.	

6.13.29.10 GeneratorControlSource

The source of controls for a generating unit.

Literal	Description	Code
unavailable	Not available.	
offAGC	Off of automatic generation control (AGC).	
onAGC	On automatic generation control (AGC).	
plantControl	Plant is controlling.	

6.13.29.11 Currency

Monetary currencies. This list is not exhaustive.

Literal	Description	Code
USD	US dollar	
EUR	European euro	
AUD	Australian dollar	
CAD	Canadian dollar	
CHF	Swiss francs	
CNY	Chinese yuan renminbi	
DKK	Danish crown	
GBP	British pound	
JPY	Japanese yen	
NOK	Norwegian crown	
RUR	Russian ruble	
SEK	Swedish crown	
INR	India rupees	
other	Another type of currency.	

6.13.29.12 HydroEnergyConversionKind

Specifies the capability of the hydro generating unit to convert energy as a generator or pump.

Literal	Description	Code
generator	Able to generate power, but not able to pump water for energy storage.	
pumpAndGenerator	Able to both generate power and pump water for energy storage.	

6.13.29.13 HydroPlantStorageKind

The type of hydro power plant.

Literal	Description	Code
runOfRiver	Run of river.	
pumpedStorage	Pumped storage.	
storage	Storage.	

6.13.29.14 WindGenUnitKind

Kind of wind generating unit.

Literal	Description	Code
offshore	The wind generating unit is located offshore.	
onshore	The wind generating unit is located onshore.	

6.13.29.15 CurveStyle

Style or shape of curve.

Literal	Description	Code
constantYValue	The Y-axis values are assumed constant until the next curve point and prior to the first curve point.	
straightLineYValues	The Y-axis values are assumed to be a straight line between values. Also known as linear interpolation.	

6.13.29.16 LimitTypeKind

The enumeration defines the kinds of the limit types.

Literal	Description	Code
patl	The Permanent Admissible Transmission Loading (PATL) is the loading in Amps, MVA or MW that can be accepted by a network branch for an unlimited duration without any risk for the material. The duration attribute is not used and shall be excluded for the PATL limit type. Hence only one limit value exists for the PATL type.	
patlt	Permanent Admissible Transmission Loading Threshold (PATLT) is a value in engineering units defined for PATL and calculated using percentage less than 100 of the PATL type intended to alert operators of an arising condition. The percentage should be given in the name of the OperationalLimitSet. The aceptableDuration is another way to express the severity of the limit.	
tatl	Temporarily Admissible Transmission Loading (TATL) which is the loading in Amps, MVA or MW that can be accepted by a branch for a certain limited duration. The TATL can be defined in different ways: – as a fixed percentage of the PATL for a given time (for example, 115 % of the PATL that can be accepted during 15 minutes), – pairs of TATL type and Duration calculated for each line taking into account its particular configuration and conditions of functioning (for example, it can define a TATL acceptable during 20 minutes and another one acceptable during 10 minutes). Such a definition of TATL can depend on the initial operating conditions of the network element (sag situation of a line). The duration attribute can be used define several TATL limit types. Hence multiple TATL limit values may exist having different durations.	
tc	Tripping Current (TC) is the ultimate intensity without any delay. It is defined as the threshold the line will trip without any possible remedial actions. The tripping of the network element is ordered by protections against short circuits or by overload protections, but in any case, the activation delay of these protections is not compatible with the reaction delay of an operator (less than one minute). The duration is always zero and the duration attribute may be left out. Hence only one limit value exists for the TC type.	
tct	Tripping Current Threshold (TCT) is a value in engineering units defined for TC and calculated using percentage less than 100 of the TC type intended to alert operators of an arising condition. The percentage should be given in the name of the OperationalLimitSet. The aceptableDuration is another way to express the severity of the limit.	
highVoltage	Referring to the rating of the equipments, a voltage too high can lead to accelerated ageing or the destruction of the equipment. This limit type may or may not have duration.	
lowVoltage	A too low voltage can disturb the normal operation of some protections and transformer equipped with on-load tap changers, electronic power devices or can affect the behaviour of the auxiliaries of generation units. This limit type may or may not have duration.	

6.13.29.17 OperationalLimitDirectionKind

The direction attribute describes the side of a limit that is a violation.

Literal	Description	Code
high	High means that a monitored value above the limit value is a violation. If applied to a terminal flow, the positive direction is into the terminal.	
low	Low means a monitored value below the limit is a violation. If applied to a terminal flow, the positive direction is into the terminal.	
absoluteValue	An absoluteValue limit means that a monitored absolute value above the limit value is a violation.	

6.13.29.18 PetersenCoilModeKind

The mode of operation for a Petersen coil.

Literal	Description	Code
fixed	Fixed position.	
manual	Manual positioning.	
automaticPositioning	Automatic positioning.	

6.13.29.19 WindingConnection

Winding connection type.

Literal	Description	Code
D	Delta	
Y	Wye	
Z	ZigZag	
Yn	Wye, with neutral brought out for grounding.	
Zn	ZigZag, with neutral brought out for grounding.	
A	Autotransformer common winding	
I	Independent winding, for single-phase connections	

6.13.29.20 TransformerControlMode

Control modes for a transformer.

Literal	Description	Code
volt	Voltage control	
reactive	Reactive power flow control	

6.13.29.21 RegulatingControlModeKind

The kind of regulation model. For example regulating voltage, reactive power, active power, etc.

Literal	Description	Code
voltage	Voltage is specified.	
activePower	Active power is specified.	
reactivePower	Reactive power is specified.	
currentFlow	Current flow is specified.	
admittance	Admittance is specified.	
timeScheduled	Control switches on/off by time of day. The times may change on the weekend, or in different seasons.	
temperature	Control switches on/off based on the local temperature (i.e., a thermostat).	
powerFactor	Power factor is specified.	

6.13.29.22 SVCControlMode

Static VAr Compensator control mode.

Literal	Description	Code
reactivePower		
voltage		

6.13.29.23 ShortCircuitRotorKind

Type of rotor, used by short circuit applications.

Literal	Description	Code
salientPole1	Salient pole 1 in theIEC 60909	
salientPole2	Salient pole 2 in IEC 60909	
turboSeries1	Turbo Series 1 in theIEC 60909	
turboSeries2	Turbo series 2 in IEC 60909	

6.13.29.24 SynchronousMachineKind

Synchronous machine type.

Literal	Description	Code
generator		
condenser		
generatorOrCondenser		
motor		
generatorOrMotor		
motorOrCondenser		
generatorOrCondenserOrMotor		

6.13.29.25 ControlAreaTypeKind

The type of control area.

Literal	Description	Code
AGC	Used for automatic generation control.	
Forecast	Used for load forecast.	
Interchange	Used for interchange specification or control.	

6.13.30 Primitives

6.13.30.1 String

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

6.13.30.2 Date

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

6.13.30.3 Float

A floating point number. The range is unspecified and not limited.

6.13.30.4 Integer

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

6.13.30.5 Float

A floating point number. The range is unspecified and not limited.

6.13.30.6 Boolean

A type with the value space "true" and "false".

6.13.30.7 DateTime

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.

6.13.30.8 Decimal

Decimal is the base-10 notational system for representing real numbers.

6.13.30.9 MonthDay

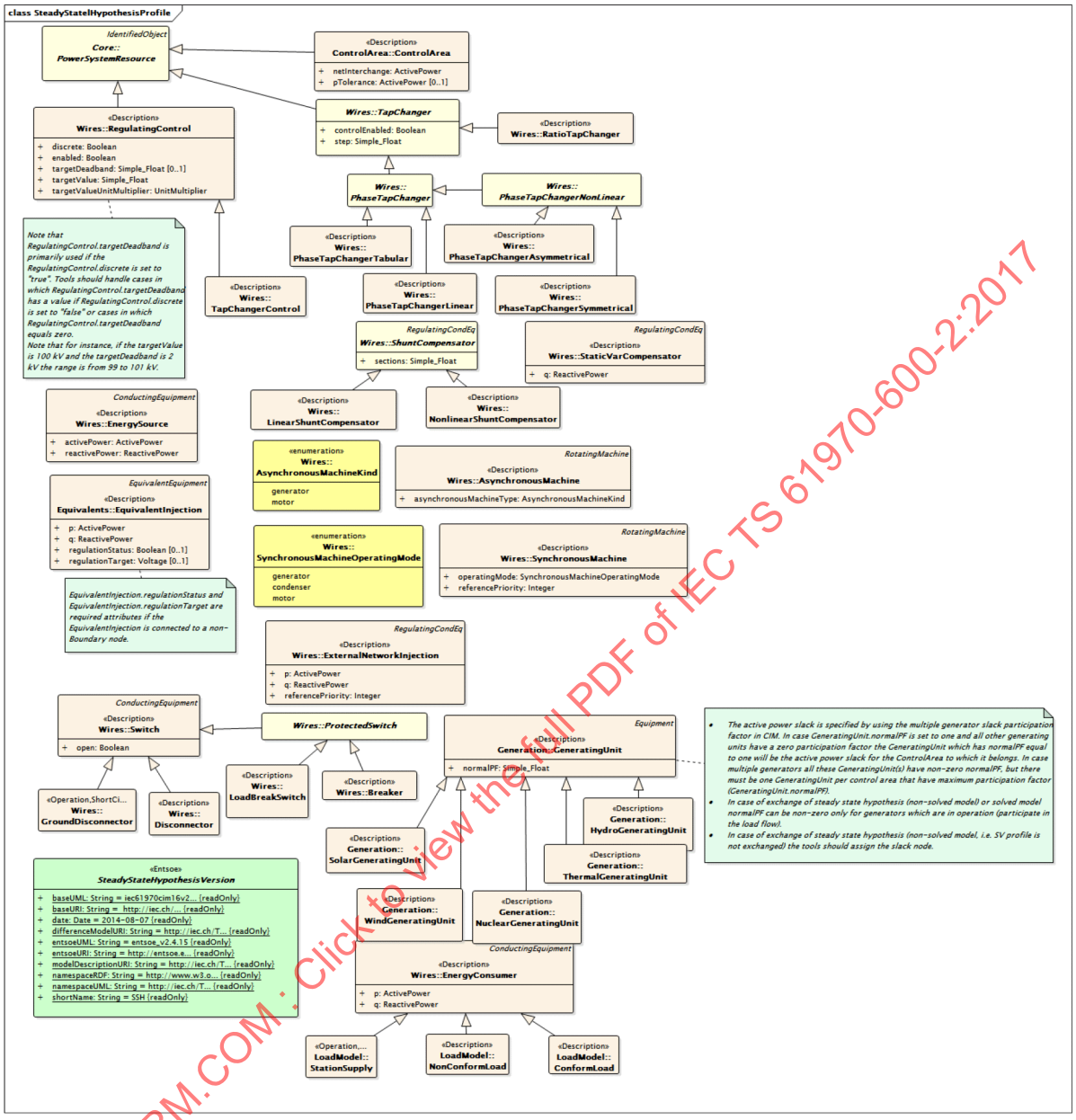
MonthDay format as "--mm-dd", which conforms to XSD data type gMonthDay.

7 SteadyStateHypothesisProfile

7.1 General

This profile has been built on the basis of IEC 61970-456 and adjusted to fit the purpose of the ENTSO-E CGMES.

Figure 28 shows the diagram.



IEC

Figure 28 – diagram SteadyStateIHypothesisProfile

7.2 (Entsoe) SteadyStateHypothesisVersion(Abstract)

Version details.

Table 357 shows all attributes of SteadyStateHypothesisVersion.

Table 357 – Attributes of SteadyStateHypothesisProfile::SteadyStateHypothesisVersion

Attribute name	mult.	Attribute type	Description
baseUML (Entsoe)	1..1	String	ConstantValue= iec61970cim16v28_iec61968cim12v08_iec62325cim03v01a Base UML provided by CIM model manager.
baseURI (Entsoe)	1..1	String	ConstantValue= http://iec.ch/TC57/2013/61970-456/SteadyStateHypothesis/1 Profile URI used in the Model Exchange header and defined in IEC standards. It uniquely identifies the Profile and its version. It is given for information only and to identify the closest IEC profile to which this CGMES profile is based on.
date (Entsoe)	1..1	Date	ConstantValue= 2014-08-07 Profile creation date Form is YYYY-MM-DD, for example for January 5, 2009 it is 2009-01-05.
differenceModelURI (Entsoe)	1..1	String	ConstantValue= http://iec.ch/TC57/61970-552/DifferenceModel/1# Difference model URI defined by IEC 61970-552.
entsoeUML (Entsoe)	1..1	String	ConstantValue= entsoe_v2.4.15 UML provided by ENTSO-E.
entsoeURI (Entsoe)	1..1	String	ConstantValue= http://entsoe.eu/CIM/SteadyStateHypothesis/1/1 Profile URI defined by ENTSO-E and used in the Model Exchange header. It uniquely identifies the Profile and its version. The last two elements in the URI (http://entsoe.eu/CIM/SteadyStateHypothesis/yy/zzz) indicate major and minor versions where: - yy – indicates a major version; - zzz – indicates a minor version.
modelDescriptionURI (Entsoe)	1..1	String	ConstantValue= http://iec.ch/TC57/61970-552/ModelDescription/1# Model Description URI defined by IEC 61970-552.
namespaceRDF (Entsoe)	1..1	String	ConstantValue= http://www.w3.org/1999/02/22-rdf-syntax-ns# RDF namespace.
namespaceUML (Entsoe)	1..1	String	ConstantValue= http://iec.ch/TC57/2013/CIM-schema-cim16# CIM UML namespace.
shortName (Entsoe)	1..1	String	ConstantValue= SSH The short name of the profile used in profile documentation.

7.3 Core

7.3.1 General

Figure 29 shows the diagram.

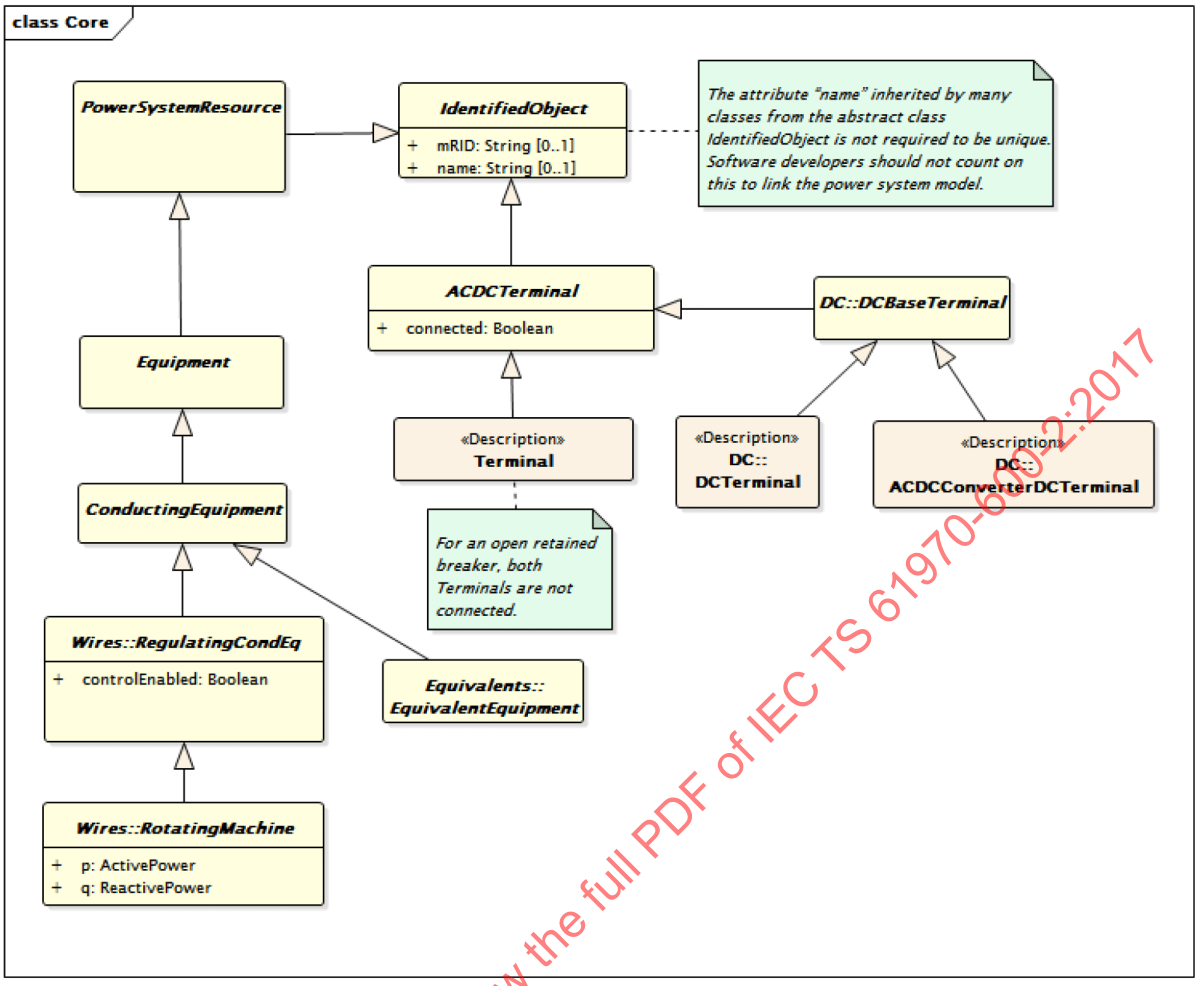


Figure 29 – diagram Core

7.3.2 ACDCTerminal(Abstract)

An electrical connection point (AC or DC) to a piece of conducting equipment. Terminals are connected at physical connection points called connectivity nodes.

Inheritance path: ->IdentifiedObject

Table 358 shows all attributes of ACDCTerminal.

Table 358 – Attributes of Core::ACDCTerminal

Attribute name	mult.	Attribute type	Description
connected	1..1	Boolean	The connected status is related to a bus-branch model and the topological node to terminal relation. True implies the terminal is connected to the related topological node and false implies it is not. In a bus-branch model, the connected status is used to tell if equipment is disconnected without having to change the connectivity described by the topological node to terminal relation. A valid case is that conducting equipment can be connected in one end and open in the other. In particular for an AC line segment, where the reactive line charging can be significant, this is a relevant case.
mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject

7.3.3 ConductingEquipment(Abstract)

The parts of the AC power system that are designed to carry current or that are conductively connected through terminals.

Inheritance path: ->Equipment->PowerSystemResource->IdentifiedObject

Table 359 shows all attributes of ConductingEquipment.

Table 359 – Attributes of Core::ConductingEquipment

Attribute name	mult.	Attribute type	Description
mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject

7.3.4 Equipment(Abstract)

The parts of a power system that are physical devices, electronic or mechanical.

Inheritance path: ->PowerSystemResource->IdentifiedObject

Table 360 shows all attributes of Equipment.

Table 360 – Attributes of Core::Equipment

Attribute name	mult.	Attribute type	Description
mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject

7.3.5 IdentifiedObject(Abstract)

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

OCL constraint:R.4.10.11. Name length restriction (optional)

The attribute “name” inherited by many classes from the abstract class IdentifiedObject is not required to be unique. Software developers should not count on this to link the power system model.

Table 361 shows all attributes of IdentifiedObject.

Table 361 – Attributes of Core::IdentifiedObject

Attribute name	mult.	Attribute type	Description
mRID	0..1	String	Master resource identifier issued by a model authority. The mRID is globally 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 61970-552:2013, the mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements.
name	0..1	String	The name is any free human readable and possibly non unique text naming the object.

7.3.6 PowerSystemResource(Abstract)

A power system resource can be an item of equipment such as a switch, an equipment container containing many individual items of equipment such as a substation, or an organisational entity such as sub-control area. Power system resources can have measurements associated.

Inheritance path: ->IdentifiedObject

Table 362 shows all attributes of PowerSystemResource.

Table 362 – Attributes of Core::PowerSystemResource

Attribute name	mult.	Attribute type	Description
mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject

7.3.7 (Description) Terminal

An AC electrical connection point to a piece of conducting equipment. Terminals are connected at physical connection points called connectivity nodes.

For an open retained breaker, both Terminals are not connected.

Inheritance path: ->ACDCTerminal->IdentifiedObject

Table 363 shows all attributes of Terminal.

Table 363 – Attributes of Core::Terminal

Attribute name	mult.	Attribute type	Description
connected	1..1	Boolean	see ACDCTerminal
mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject

7.4 ControlArea

7.4.1 (Description) ControlArea

A control area is a grouping of generating units and/or loads and a cutset of tie lines (as terminals) which may be used for a variety of purposes including automatic generation control, powerflow solution area interchange control specification, and input to load forecasting. Note that any number of overlapping control area specifications can be superimposed on the physical model.

Inheritance path: ->PowerSystemResource->IdentifiedObject

Table 364 shows all attributes of ControlArea.

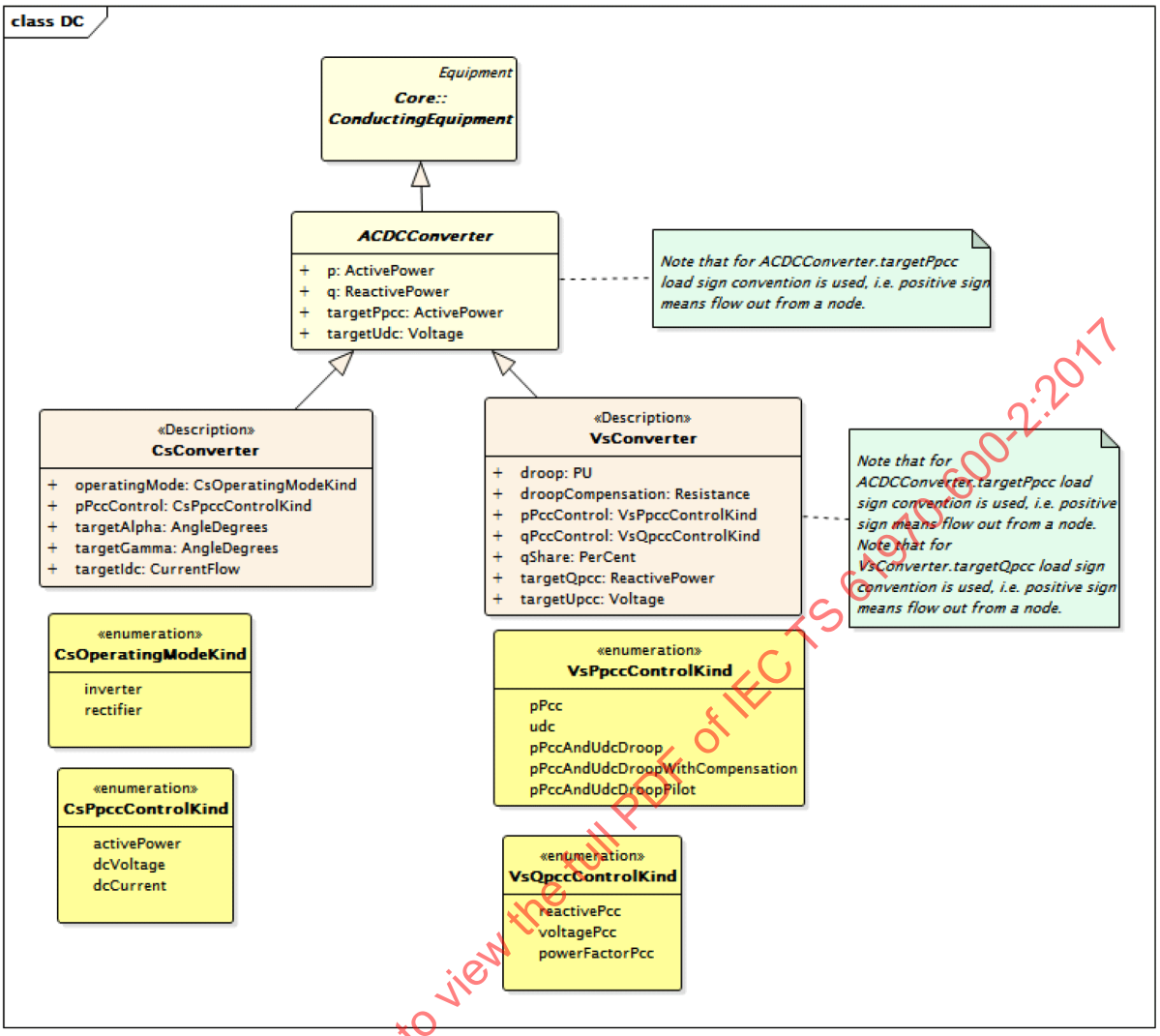
Table 364 – Attributes of ControlArea::ControlArea

Attribute name	mult.	Attribute type	Description
netInterchange (Description)	1..1	ActivePower	The specified positive net interchange into the control area, i.e. positive sign means flow in to the area.
pTolerance (Description)	0..1	ActivePower	Active power net interchange tolerance
mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject

7.5 DC

7.5.1 General

Figure 30 shows the diagram.



IEC

Figure 30 – diagram DC

7.5.2 ACDCConverter(Abstract)

A unit with valves for three phases, together with unit control equipment, essential protective and switching devices, DC storage capacitors, phase reactors and auxiliaries, if any, used for conversion.

Note that for ACDCConverter.targetPpcc load sign convention is used, i.e. positive sign means flow out from a node.

Inheritance path: ->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 365 shows all attributes of ACDCConverter.

Table 365 – Attributes of DC::ACDCConverter

Attribute name	mult.	Attribute type	Description
p	1..1	ActivePower	Active power at the point of common coupling. Load sign convention is used, i.e. positive sign means flow out from a node. Starting value for a steady state solution in the case a simplified power flow model is used.
q	1..1	ReactivePower	Reactive power at the point of common coupling. Load sign convention is used, i.e. positive sign means flow out from a node. Starting value for a steady state solution in the case a simplified power flow model is used.
targetPpcc	1..1	ActivePower	Real power injection target in AC grid, at point of common coupling.
targetUdc	1..1	Voltage	Target value for DC voltage magnitude.
mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject

7.5.3 (Description) ACDCConverterDCTerminal

A DC electrical connection point at the AC/DC converter. The AC/DC converter is electrically connected also to the AC side. The AC connection is inherited from the AC conducting equipment in the same way as any other AC equipment. The AC/DC converter DC terminal is separate from generic DC terminal to restrict the connection with the AC side to AC/DC converter and so that no other DC conducting equipment can be connected to the AC side.

Inheritance path: ->DCBaseTerminal->ACDCTerminal->IdentifiedObject

Table 366 shows all attributes of ACDCConverterDCTerminal.

Table 366 – Attributes of DC::ACDCConverterDCTerminal

Attribute name	mult.	Attribute type	Description
connected	1..1	Boolean	see ACDCTerminal
mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject

7.5.4 (Description) CsConverter

DC side of the current source converter (CSC).

Inheritance path: ->ACDCConverter->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 367 shows all attributes of CsConverter.

Table 367 – Attributes of DC::CsConverter

Attribute name	mult.	Attribute type	Description
operatingMode (Description)	1..1	CsOperatingModeKind	Indicates whether the DC pole is operating as an inverter or as a rectifier. CSC control variable used in power flow.
pPccControl (Description)	1..1	CsPpccControlKind	
targetAlpha (Description)	1..1	AngleDegrees	Target firing angle. CSC control variable used in power flow.
targetGamma (Description)	1..1	AngleDegrees	Target extinction angle. CSC control variable used in power flow.
targetIdc (Description)	1..1	CurrentFlow	DC current target value. CSC control variable used in power flow.
p	1..1	ActivePower	see ACDCConverter
q	1..1	ReactivePower	see ACDCConverter
targetPpcc	1..1	ActivePower	see ACDCConverter
targetUdc	1..1	Voltage	see ACDCConverter
mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject

7.5.5 DCBaseTerminal(Abstract)

An electrical connection point at a piece of DC conducting equipment. DC terminals are connected at one physical DC node that may have multiple DC terminals connected. A DC node is similar to an AC connectivity node. The model enforces that DC connections are distinct from AC connections.

Inheritance path: ->ACDCTerminal->IdentifiedObject

Table 368 shows all attributes of DCBaseTerminal.

Table 368 – Attributes of DC::DCBaseTerminal

Attribute name	mult.	Attribute type	Description
connected	1..1	Boolean	see ACDCTerminal
mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject

7.5.6 (Description) DCTerminal

An electrical connection point to generic DC conducting equipment.

Inheritance path: ->DCBaseTerminal->ACDCTerminal->IdentifiedObject

Table 369 shows all attributes of DCTerminal.

Table 369 – Attributes of DC::DCTerminal

Attribute name	mult.	Attribute type	Description
connected	1..1	Boolean	see ACDCTerminal
mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject

7.5.7 (Description) VsConverter

DC side of the voltage source converter (VSC).

Note that for ACDCConverter.targetPpcc load sign convention is used, i.e. positive sign means flow out from a node. Note that for VsConverter.targetQpcc load sign convention is used, i.e. positive sign means flow out from a node.

Inheritance path: ->ACDCConverter->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 370 shows all attributes of VsConverter.

Table 370 – Attributes of DC::VsConverter

Attribute name	mult.	Attribute type	Description
droop (Description)	1..1	PU	Droop constant; pu value is obtained as D [kV^2 / MW] x Sb / Ubdc^2.
droopCompensation (Description)	1..1	Resistance	Compensation (resistance) constant. Used to compensate for voltage drop when controlling voltage at a distant bus.
pPccControl (Description)	1..1	VsPpccControlKind	Kind of control of real power and/or DC voltage.
qPccControl (Description)	1..1	VsQpccControlKind	
qShare (Description)	1..1	PerCent	Reactive power sharing factor among parallel converters on Uac control.
targetQpcc (Description)	1..1	ReactivePower	Reactive power injection target in AC grid, at point of common coupling.
targetUpcc (Description)	1..1	Voltage	Voltage target in AC grid, at point of common coupling.
p	1..1	ActivePower	see ACDCConverter
q	1..1	ReactivePower	see ACDCConverter
targetPpcc	1..1	ActivePower	see ACDCConverter
targetUdc	1..1	Voltage	see ACDCConverter
mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject

7.5.8 Enumerations

7.5.8.1 CsOperatingModeKind

Operating mode for HVDC line operating as Current Source Converter.

Literal	Description	Code
inverter	Operating as inverter	
rectifier	Operating as rectifier.	

7.5.8.2 CsPpccControlKind

Active power control modes for HVDC line operating as Current Source Converter.

Literal	Description	Code
activePower	Active power control at AC side.	
dcVoltage	DC voltage control.	
dcCurrent	DC current control	

7.5.8.3 VsPpccControlKind

Types applicable to the control of real power and/or DC voltage by voltage source converter.

Literal	Description	Code
pPcc	Control variable (target) is real power at PCC bus.	
udc	Control variable (target) is DC voltage and real power at PCC bus is derived.	
pPccAndUdcDroop	Control variables (targets) are both active power at point of common coupling and local DC voltage, with the droop.	
pPccAndUdcDroopWithCompensation	Control variables (targets) are both active power at point of common coupling and compensated DC voltage, with the droop; compensation factor is the resistance, as an approximation of the DC voltage of a common (real or virtual) node in the DC network.	
pPccAndUdcDroopPilot	Control variables (targets) are both active power at point of common coupling and the pilot DC voltage, with the droop.	

7.5.8.4 VsQpccControlKind

Literal	Description	Code
reactivePcc		
voltagePcc		
powerFactorPcc		

7.6 Equivalents

7.6.1 EquivalentEquipment(Abstract)

The class represents equivalent objects that are the result of a network reduction. The class is the base for equivalent objects of different types.

Inheritance path: ->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 371 shows all attributes of EquivalentEquipment.

Table 371 – Attributes of Equivalents::EquivalentEquipment

Attribute name	mult.	Attribute type	Description
mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject

7.6.2 (Description) EquivalentInjection

This class represents equivalent injections (generation or load). Voltage regulation is allowed only at the point of connection.

EquivalentInjection.regulationStatus and EquivalentInjection.regulationTarget are required attributes if the EquivalentInjection is connected to a non-Boundary node.

Inheritance path: ->EquivalentEquipment->ConductingEquipment->Equipment->PowerSystemResource->IdentifiedObject

Table 372 shows all attributes of EquivalentInjection.

Table 372 – Attributes of Equivalents::EquivalentInjection

Attribute name	mult.	Attribute type	Description
regulationStatus (Description)	0..1	Boolean	Specifies the default regulation status of the EquivalentInjection. True is regulating. False is not regulating.
regulationTarget (Description)	0..1	Voltage	The target voltage for voltage regulation.
p (Description)	1..1	ActivePower	Equivalent active power injection. Load sign convention is used, i.e. positive sign means flow out from a node. Starting value for steady state solutions.
q (Description)	1..1	ReactivePower	Equivalent reactive power injection. Load sign convention is used, i.e. positive sign means flow out from a node. Starting value for steady state solutions.
mRID	0..1	String	see IdentifiedObject
name	0..1	String	see IdentifiedObject

7.7 Generation

7.7.1 (Description) GeneratingUnit

A single or set of synchronous machines for converting mechanical power into alternating-current power. For example, individual machines within a set may be defined for scheduling purposes while a single control signal is derived for the set. In this case there would be a GeneratingUnit for each member of the set and an additional GeneratingUnit corresponding to the set.

- The active power slack is specified by using the multiple generator slack participation factor in CIM. In case GeneratingUnit.normalPF is set to one and all other generating units have a zero participation factor the GeneratingUnit which has normalPF equal to one will be the active power slack for the ControlArea to which it belongs. In case of multiple generators all these GeneratingUnit(s) have non-zero normalPF, but there must be one GeneratingUnit per control area that have maximum participation factor (GeneratingUnit.normalPF).