LTDS CIM profiles

Edition 1.00 Draft

19 Aug 2022

**Document history**

Any person intervening in the present document is invited to complete the table below before sending the document elsewhere. The purpose is to allow all actors to see all changes introduced and the intervening persons.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name of intervening person | Document received | | Brief description of the  changes introduced | Document sent | |
| From | Date | To | Date |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

CONTENTS

FOREWORD 21

INTRODUCTION 21

1 Scope 22

2 Normative references 22

3 Terms and definitions 22

4 Profile specification 22

4.1 General 22

4.2 Version information 24

4.2.1 LTDS Equipment profile 24

4.2.2 LTDS Short circuit profile 24

4.3 Requirements and constraints definition 24

5 Overview 25

6 Use cases 25

6.1 General 25

6.2 xxx 25

7 Architecture and usage 25

7.1 General 25

7.2 xxx profile 25

7.3 xxx profile 26

7.4 Geograpfical location profile 26

7.5 Diagram layout profile 26

7.6 Instance file (distribution) packaging and dependency 26

8 LTDS Equipment profile 27

8.1 Detailed specification 27

8.1.1 General 27

8.1.2 (abstract) ACDCTerminal 44

8.1.3 ACLineSegment 45

8.1.4 ActivePowerLimit 46

8.1.5 ApparentPowerLimit 46

8.1.6 AsynchronousMachine 46

8.1.7 (abstract) AuxiliaryEquipment 47

8.1.8 BaseVoltage 47

8.1.9 (abstract) BasicIntervalSchedule 48

8.1.10 BatteryUnit 48

8.1.11 Bay 48

8.1.12 (European) BoundaryPoint 49

8.1.13 Breaker 50

8.1.14 BusbarSection 51

8.1.15 BusNameMarker 51

8.1.16 CAESPlant 52

8.1.17 Clamp 52

8.1.18 ConformLoadSchedule 52

8.1.19 CogenerationPlant 53

8.1.20 CombinedCyclePlant 53

8.1.21 (abstract) ConductingEquipment 54

8.1.22 (abstract) Conductor 54

8.1.23 ConformLoad 54

8.1.24 ConformLoadGroup 55

8.1.25 ConnectivityNode 55

8.1.26 (abstract) ConnectivityNodeContainer 56

8.1.27 (abstract) Connector 56

8.1.28 ControlArea 56

8.1.29 ControlAreaGeneratingUnit 57

8.1.30 CurrentLimit 58

8.1.31 CurrentTransformer 58

8.1.32 (abstract) Curve 59

8.1.33 CurveData root class 59

8.1.34 Cut 59

8.1.35 DayType 60

8.1.36 RegulationSchedule 60

8.1.37 Disconnector 61

8.1.38 DisconnectingCircuitBreaker 61

8.1.39 (abstract) EarthFaultCompensator 62

8.1.40 (abstract) EnergyArea 62

8.1.41 (abstract) EnergyConnection 62

8.1.42 EnergyConsumer 63

8.1.43 EnergySchedulingType 64

8.1.44 TapSchedule 64

8.1.45 EnergySource 64

8.1.46 (abstract) Equipment 65

8.1.47 (abstract) EquipmentContainer 66

8.1.48 EquivalentBranch 66

8.1.49 (abstract) EquivalentEquipment 67

8.1.50 EquivalentInjection 67

8.1.51 EquivalentNetwork 68

8.1.52 EquivalentShunt 68

8.1.53 ExternalNetworkInjection 69

8.1.54 FaultIndicator 69

8.1.55 FossilFuel 70

8.1.56 Fuse 70

8.1.57 GeneratingUnit 71

8.1.58 GeographicalRegion 72

8.1.59 GrossToNetActivePowerCurve 72

8.1.60 Ground 73

8.1.61 GroundDisconnector 73

8.1.62 GroundingImpedance 74

8.1.63 HydroGeneratingUnit 74

8.1.64 HydroPowerPlant 75

8.1.65 HydroPump 75

8.1.66 (abstract) IdentifiedObject root class 76

8.1.67 Jumper 76

8.1.68 Junction 77

8.1.69 Line 77

8.1.70 LinearShuntCompensator 77

8.1.71 LoadArea 78

8.1.72 LoadBreakSwitch 78

8.1.73 (abstract) LoadGroup 79

8.1.74 LoadResponseCharacteristic 79

8.1.75 NonConformLoad 80

8.1.76 NonConformLoadGroup 81

8.1.77 NonConformLoadSchedule 81

8.1.78 NonlinearShuntCompensator 82

8.1.79 NonlinearShuntCompensatorPoint root class 82

8.1.80 NuclearGeneratingUnit 83

8.1.81 (abstract) OperationalLimit 84

8.1.82 OperationalLimitSet 84

8.1.83 OperationalLimitType 85

8.1.84 PetersenCoil 85

8.1.85 (abstract) PhaseTapChanger 86

8.1.86 PhaseTapChangerAsymmetrical 86

8.1.87 PhaseTapChangerLinear 87

8.1.88 (abstract) PhaseTapChangerNonLinear 88

8.1.89 PhaseTapChangerSymmetrical 89

8.1.90 PhaseTapChangerTable 89

8.1.91 PhaseTapChangerTablePoint 90

8.1.92 PhaseTapChangerTabular 90

8.1.93 PhotoVoltaicUnit 91

8.1.94 PostLineSensor 91

8.1.95 PotentialTransformer 91

8.1.96 PowerElectronicsConnection 92

8.1.97 (abstract) PowerElectronicsUnit 93

8.1.98 PowerElectronicsWindUnit 93

8.1.99 (abstract) PowerSystemResource 94

8.1.100 PowerTransformer 94

8.1.101 PowerTransformerEnd 94

8.1.102 (abstract) ProtectedSwitch 95

8.1.103 RatioTapChanger 96

8.1.104 RatioTapChangerTable 97

8.1.105 RatioTapChangerTablePoint 97

8.1.106 ReactiveCapabilityCurve 97

8.1.107 (abstract) RegulatingCondEq 98

8.1.108 RegulatingControl 98

8.1.109 RegularTimePoint root class 99

8.1.110 (abstract) RegularIntervalSchedule 99

8.1.111 ReportingGroup 100

8.1.112 (abstract) RotatingMachine 100

8.1.113 Season 101

8.1.114 (abstract) Sensor 101

8.1.115 (abstract) SeasonDayTypeSchedule 101

8.1.116 SeriesCompensator 102

8.1.117 (abstract) ShuntCompensator 102

8.1.118 SolarGeneratingUnit 103

8.1.119 (European) SolarPowerPlant 104

8.1.120 StaticVarCompensator 104

8.1.121 StationSupply 105

8.1.122 SubGeographicalRegion 105

8.1.123 SubLoadArea 106

8.1.124 Substation 106

8.1.125 SurgeArrester 107

8.1.126 Switch 107

8.1.127 SwitchSchedule 108

8.1.128 SynchronousMachine 108

8.1.129 (abstract) TapChanger 109

8.1.130 TapChangerControl 110

8.1.131 (abstract) TapChangerTablePoint root class 110

8.1.132 Terminal 111

8.1.133 ThermalGeneratingUnit 112

8.1.134 TieFlow 112

8.1.135 (abstract) TransformerEnd 113

8.1.136 VoltageLevel 114

8.1.137 VoltageLimit 114

8.1.138 WaveTrap 115

8.1.139 WindGeneratingUnit 115

8.1.140 (European) WindPowerPlant 116

8.1.141 ControlAreaTypeKind enumeration 116

8.1.142 Currency enumeration 116

8.1.143 CurveStyle enumeration 120

8.1.144 FuelType enumeration 120

8.1.145 GeneratorControlSource enumeration 120

8.1.146 HydroEnergyConversionKind enumeration 120

8.1.147 HydroPlantStorageKind enumeration 121

8.1.148 HydroTurbineKind enumeration 121

8.1.149 (European) LimitKind enumeration 121

8.1.150 OperationalLimitDirectionKind enumeration 122

8.1.151 PhaseCode enumeration 123

8.1.152 RegulatingControlModeKind enumeration 123

8.1.153 SynchronousMachineKind enumeration 124

8.1.154 UnitMultiplier enumeration 124

8.1.155 UnitSymbol enumeration 125

8.1.156 WindGenUnitKind enumeration 129

8.1.157 WindingConnection enumeration 130

8.1.158 ActivePower datatype 130

8.1.159 ActivePowerPerFrequency datatype 130

8.1.160 AngleDegrees datatype 130

8.1.161 ReactancePerLength datatype 131

8.1.162 ConductancePerLength datatype 131

8.1.163 ResistancePerLength datatype 131

8.1.164 SusceptancePerLength datatype 131

8.1.165 ApparentPower datatype 131

8.1.166 Conductance datatype 132

8.1.167 CurrentFlow datatype 132

8.1.168 Frequency datatype 132

8.1.169 Length datatype 132

8.1.170 Money datatype 132

8.1.171 PerCent datatype 133

8.1.172 Reactance datatype 133

8.1.173 ReactivePower datatype 133

8.1.174 RealEnergy datatype 133

8.1.175 Resistance datatype 133

8.1.176 RotationSpeed datatype 134

8.1.177 Seconds datatype 134

8.1.178 Susceptance datatype 134

8.1.179 Voltage datatype 134

8.1.180 VoltagePerReactivePower datatype 134

8.1.181 (profcim) IRI primitive 135

8.1.182 Boolean primitive 135

8.1.183 (profcim) StringIRI primitive 135

8.1.184 (profcim) StringFixedLanguage primitive 135

8.1.185 (profcim) URL primitive 135

8.1.186 Date primitive 135

8.1.187 Decimal primitive 135

8.1.188 DateTime primitive 135

8.1.189 Float primitive 135

8.1.190 Integer primitive 135

8.1.191 MonthDay primitive 136

8.1.192 String primitive 136

8.1.193 Package AdditionalClasses 136

8.2 Requirements and constraints 137

9 LTDS Short circuit profile 137

9.1 Detailed specification 137

9.1.1 General 137

9.1.2 (abstract) ACDCTerminal 140

9.1.3 (Description) ACLineSegment 140

9.1.4 (Description) AsynchronousMachine 140

9.1.5 (Description) BusbarSection 141

9.1.6 (abstract) ConductingEquipment 141

9.1.7 (abstract) Conductor 141

9.1.8 (abstract) Connector 142

9.1.9 (abstract) EarthFaultCompensator 142

9.1.10 (abstract) EnergyConnection 142

9.1.11 (Description) EnergySource 142

9.1.12 (abstract) Equipment 142

9.1.13 (Description) EquivalentBranch 143

9.1.14 (abstract) EquivalentEquipment 144

9.1.15 (Description) EquivalentInjection 144

9.1.16 (Description) ExternalNetworkInjection 145

9.1.17 (Description) GroundingImpedance 146

9.1.18 (abstract) IdentifiedObject root class 146

9.1.19 (Description) LinearShuntCompensator 146

9.1.20 MutualCoupling 146

9.1.21 (Description) NonlinearShuntCompensatorPoint root class 147

9.1.22 (Description) PetersenCoil 147

9.1.23 (abstract) PowerSystemResource 148

9.1.24 (Description) PowerTransformer 148

9.1.25 (Description) PowerTransformerEnd 149

9.1.26 (abstract) RegulatingCondEq 150

9.1.27 (abstract) RotatingMachine 150

9.1.28 (Description) SeriesCompensator 150

9.1.29 (abstract) ShuntCompensator 151

9.1.30 (Description) SynchronousMachine 151

9.1.31 (abstract) Terminal 152

9.1.32 (abstract) TransformerEnd 152

9.1.33 PetersenCoilModeKind enumeration 153

9.1.34 ShortCircuitRotorKind enumeration 153

9.1.35 UnitMultiplier enumeration 153

9.1.36 UnitSymbol enumeration 154

9.1.37 ReactancePerLength datatype 158

9.1.38 ActivePower datatype 159

9.1.39 ResistancePerLength datatype 159

9.1.40 SusceptancePerLength datatype 159

9.1.41 ConductancePerLength datatype 159

9.1.42 AngleDegrees datatype 159

9.1.43 Conductance datatype 160

9.1.44 CurrentFlow datatype 160

9.1.45 Length datatype 160

9.1.46 PerCent datatype 160

9.1.47 PU datatype 160

9.1.48 Reactance datatype 161

9.1.49 Resistance datatype 161

9.1.50 Susceptance datatype 161

9.1.51 Temperature datatype 161

9.1.52 Voltage datatype 161

9.1.53 Boolean primitive 162

9.1.54 Date primitive 162

9.1.55 DateTime primitive 162

9.1.56 Float primitive 162

9.1.57 Integer primitive 162

9.1.58 String primitive 162

9.1.59 (profcim) IRI primitive 162

9.1.60 (profcim) StringIRI primitive 162

9.1.61 (profcim) StringFixedLanguage primitive 163

9.1.62 (profcim) URL primitive 163

9.1.63 Package AdditionalClasses 163

9.2 Requirements and constraints 163

10 Geographical location profile 163

10.1 Package GeographicalLocationProfile 163

10.1.1 General 163

10.1.2 (abstract) IdentifiedObject root class 165

10.1.3 ServiceLocation 165

10.1.4 (abstract) WorkLocation 165

10.1.5 CoordinateSystem 166

10.1.6 Location 166

10.1.7 PositionPoint root class 167

10.1.8 (abstract) PowerSystemResource 167

10.1.9 StreetAddress compound 167

10.1.10 TownDetail compound 168

10.1.11 Status compound 168

10.1.12 StreetDetail compound 168

10.1.13 DateTime primitive 169

10.1.14 Boolean primitive 169

10.1.15 Integer primitive 169

10.1.16 String primitive 169

10.1.17 Date primitive 169

11 Diagram Layout profile 169

11.1 Package DiagramLayoutProfile 169

11.1.1 General 169

11.1.2 Diagram 171

11.1.3 DiagramObject 171

11.1.4 DiagramObjectGluePoint root class 172

11.1.5 DiagramObjectPoint root class 172

11.1.6 DiagramObjectStyle 173

11.1.7 DiagramStyle 173

11.1.8 (abstract) IdentifiedObject root class 173

11.1.9 TextDiagramObject 174

11.1.10 VisibilityLayer 174

11.1.11 OrientationKind enumeration 175

11.1.12 UnitMultiplier enumeration 175

11.1.13 UnitSymbol enumeration 176

11.1.14 AngleDegrees datatype 181

11.1.15 Boolean primitive 181

11.1.16 Date primitive 181

11.1.17 Float primitive 181

11.1.18 Integer primitive 181

11.1.19 String primitive 181

12 Common requirements and constraints 181

Annex A (normative) xxx 183

Annex B (informative) Examples using IEC 61970-552 serialisation (placeholder in case we want to include something like this) 184

Bibliography (to be updated) 185

Ofgem

\_\_\_\_\_\_\_\_\_\_\_\_

LTDS

Profiles

INTRODUCTION

XYZ.

# Normative references

The following referenced documents are indispensable for the application 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.

Example:

IEC 61970-301:2021, Energy management system application program interface (EMS-API) – Part 301: Common information model (CIM) base

IEC 61970-302[[1]](#footnote-2), Energy management system application program interface (EMS-API) – Part 302: Common information model (CIM) dynamics

IEC 61970-452:2021, Energy management system application program interface (EMS-API) – Part 452: CIM static transmission network model profiles

IEC 61970-456:2021, Energy management system application program interface (EMS-API) – Part 456: Solved power system state profiles

# Terms and definitions

* xyz

# LTDS Equipment profile

## Detailed specification

### General

This is the LTDS core equipment profile.

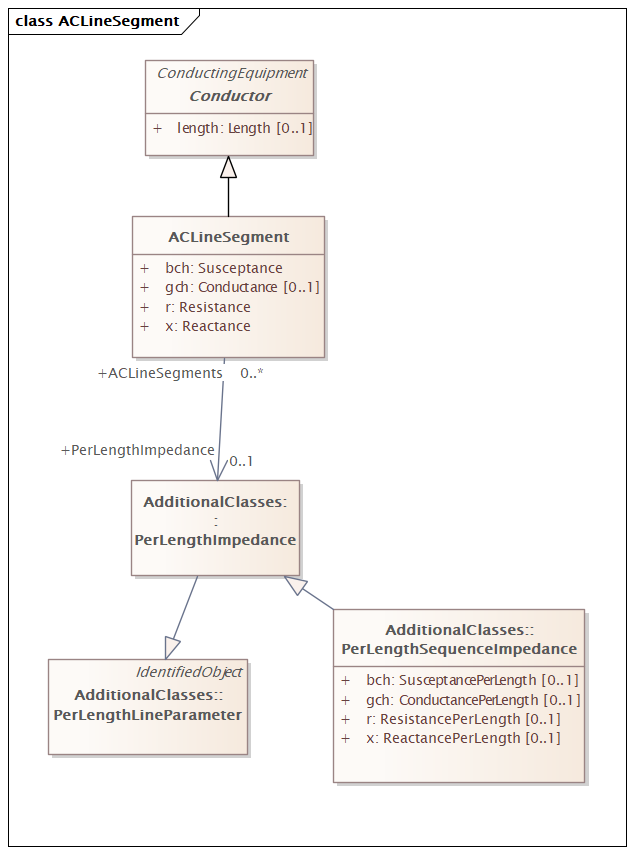


Figure 3 – Class diagram LTDSEquipmentProfile::ACLineSegment

Figure 3:

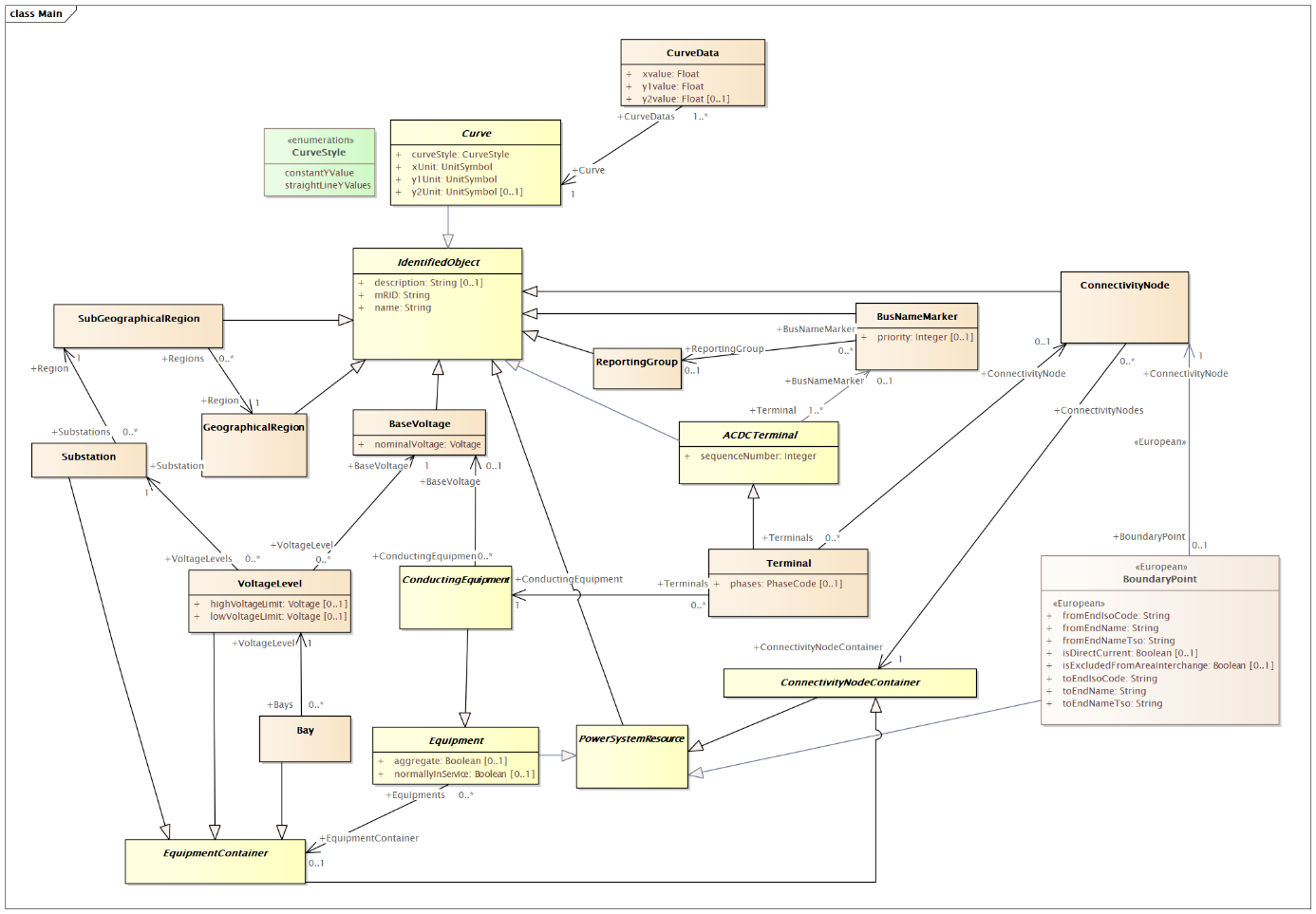


Figure 4 – Class diagram LTDSEquipmentProfile::Main

Figure 4: The main diagram of the equipment core profile.

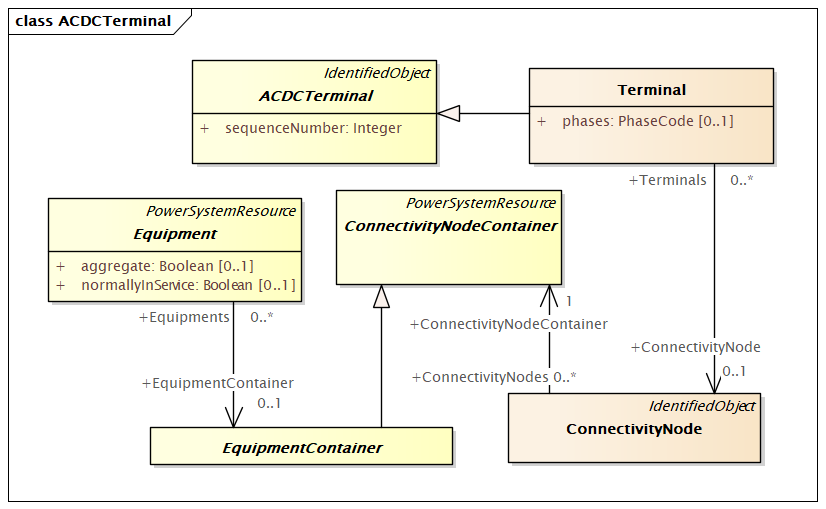


Figure 5 – Class diagram LTDSEquipmentProfile::ACDCTerminal

Figure 5: The diagram shows the ACDCTerminal and related classes.

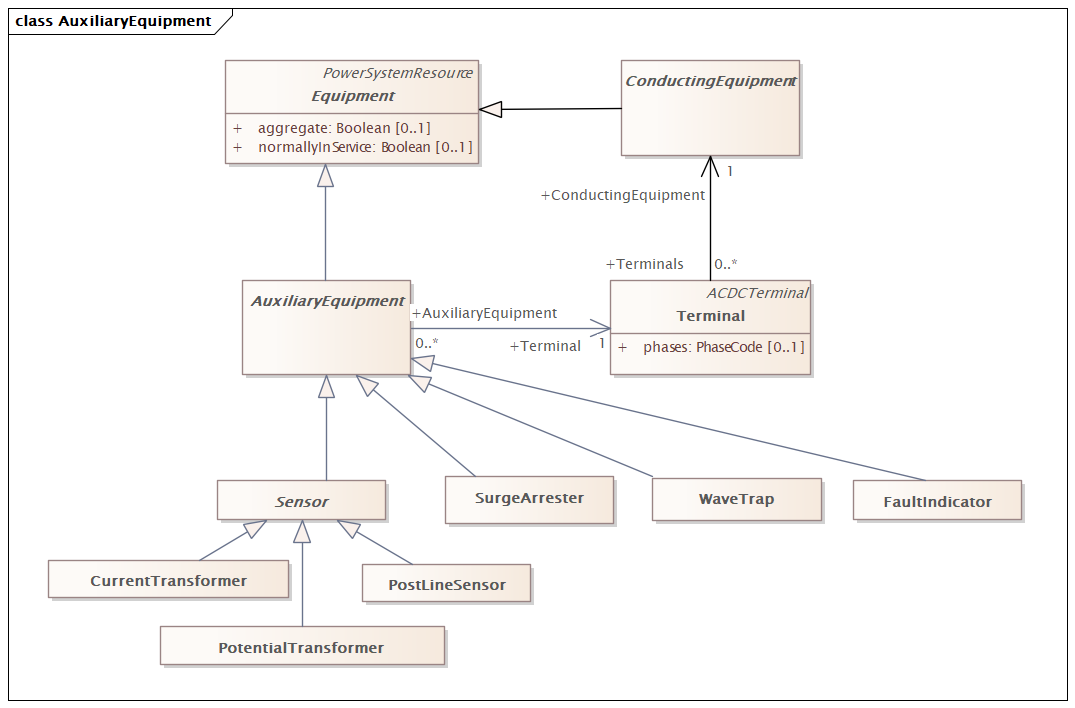


Figure 6 – Class diagram LTDSEquipmentProfile::AuxiliaryEquipment

Figure 6: The diagram shows main classes related to auxiliary equipment.

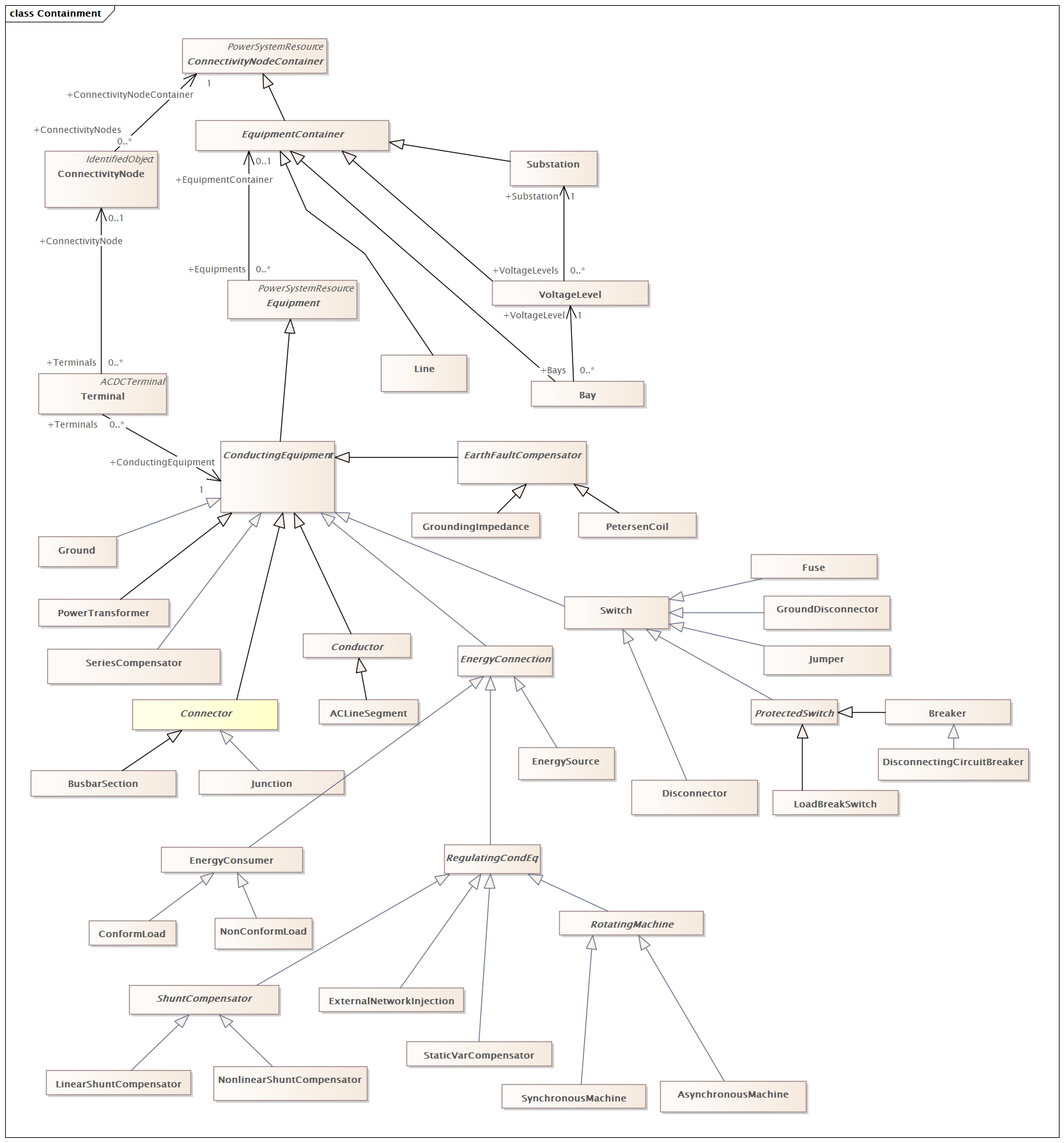


Figure 7 – Class diagram LTDSEquipmentProfile::Containment

Figure 7: The diagram shows the main classes related to containment.

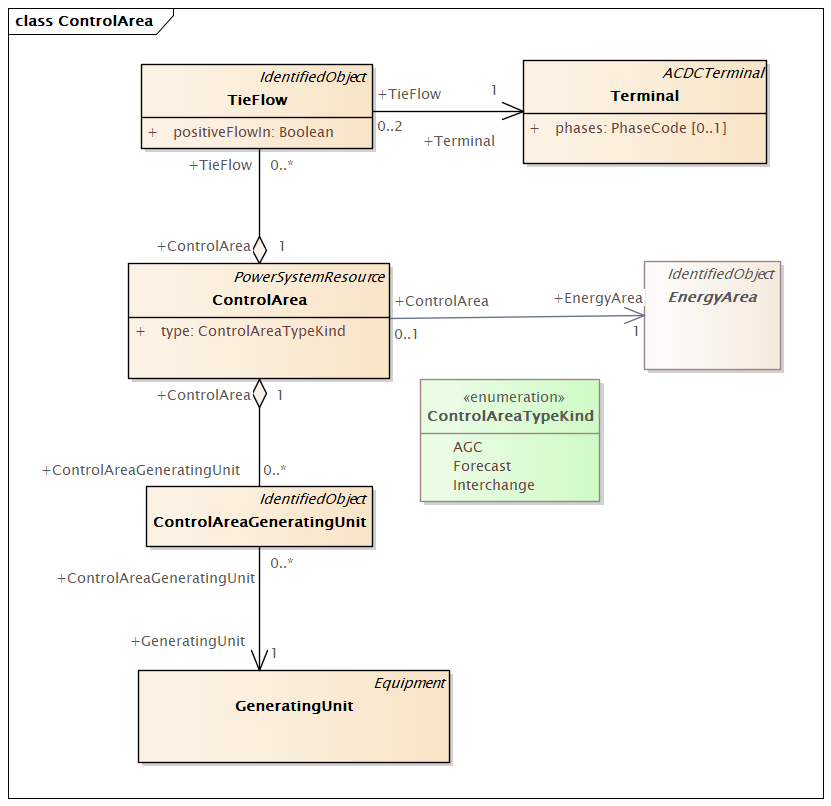


Figure 8 – Class diagram LTDSEquipmentProfile::ControlArea

Figure 8: This diagram shows control area specification and some related classes. The Terminal to AnalogValue linkages are shown for clarity in understanding the control area specification. The GeneratingUnit to Terminal linkages are also shown to illustrate how generation flows are specifically tied to the network.

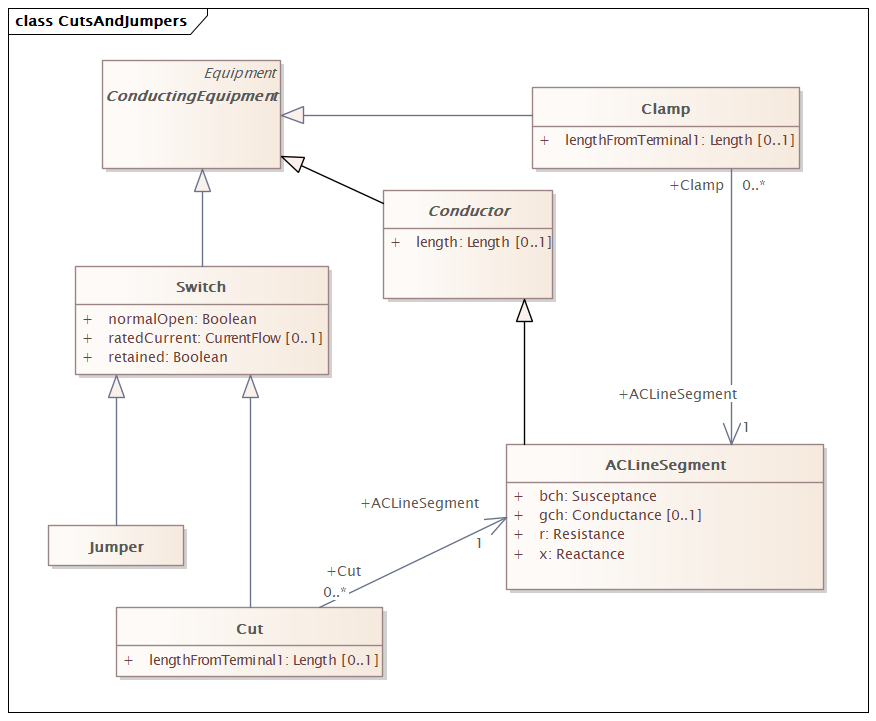


Figure 9 – Class diagram LTDSEquipmentProfile::CutsAndJumpers

Figure 9: The diagram shows classes related to cuts and jumpers.

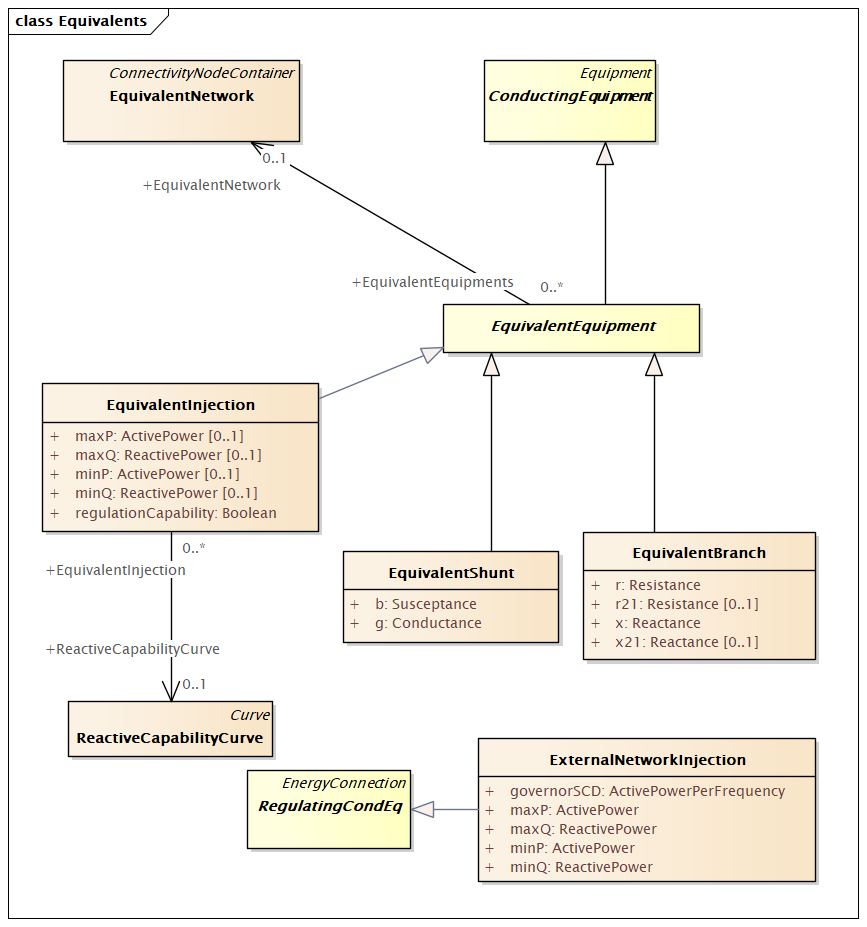


Figure 10 – Class diagram LTDSEquipmentProfile::Equivalents

Figure 10: The diagram shows classes related to equivalents.

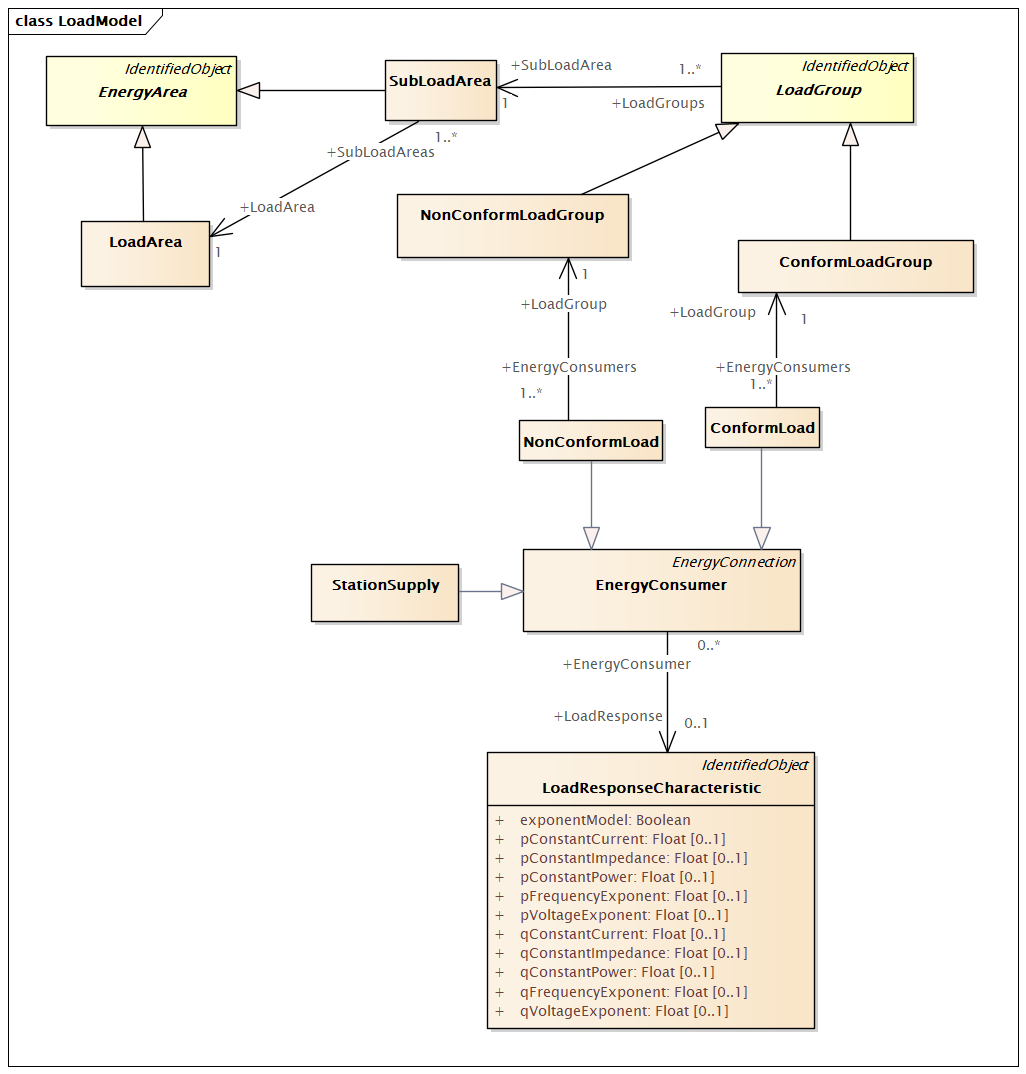


Figure 11 – Class diagram LTDSEquipmentProfile::LoadModel

Figure 11: The diagram shows classes related to the load model.

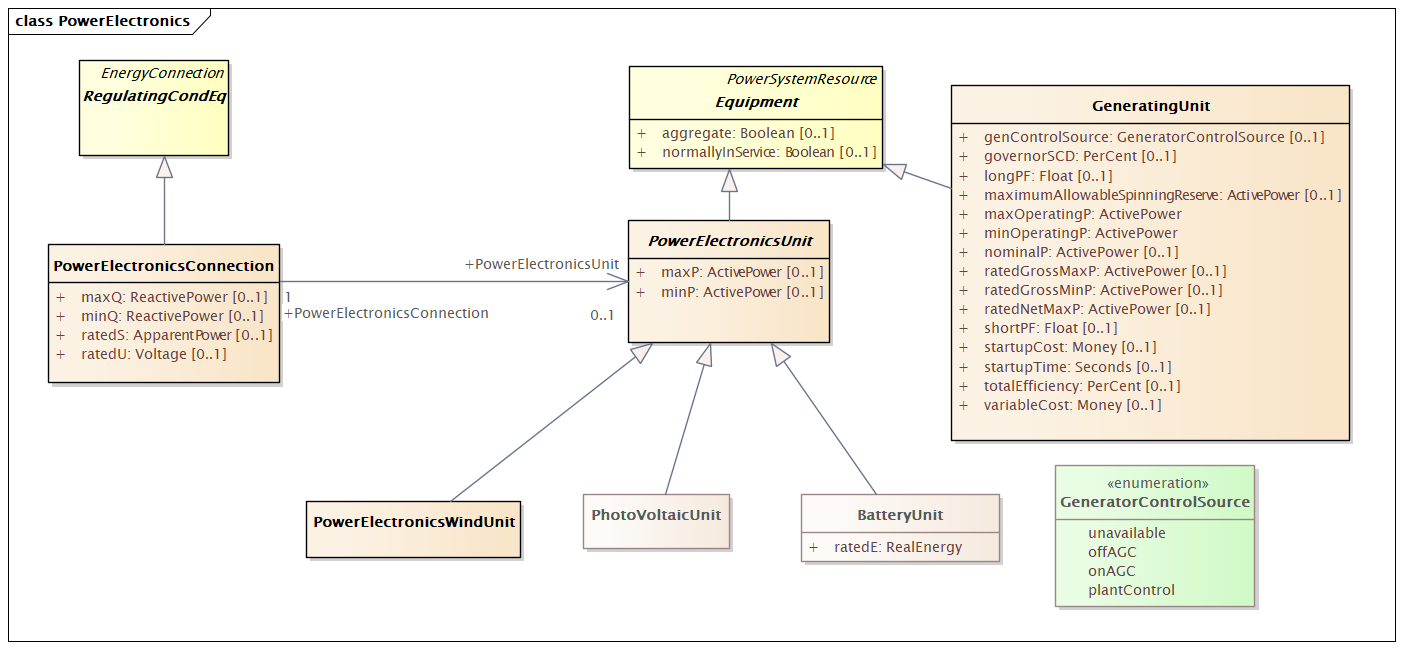


Figure 12 – Class diagram LTDSEquipmentProfile::PowerElectronics

Figure 12: This diagram shows power electronics classes and inheritance.

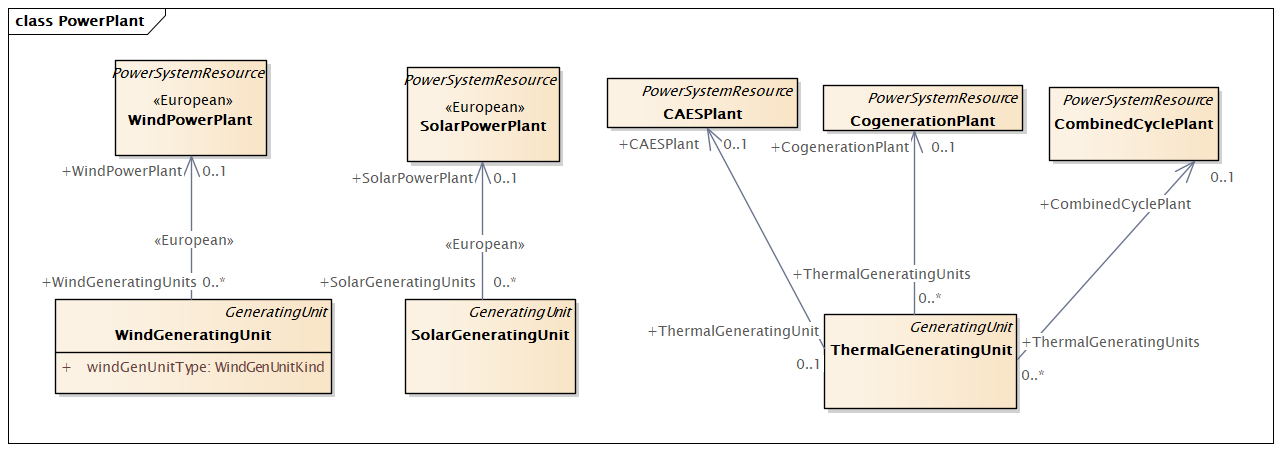


Figure 13 – Class diagram LTDSEquipmentProfile::PowerPlant

Figure 13: The diagram shows the power plant model.

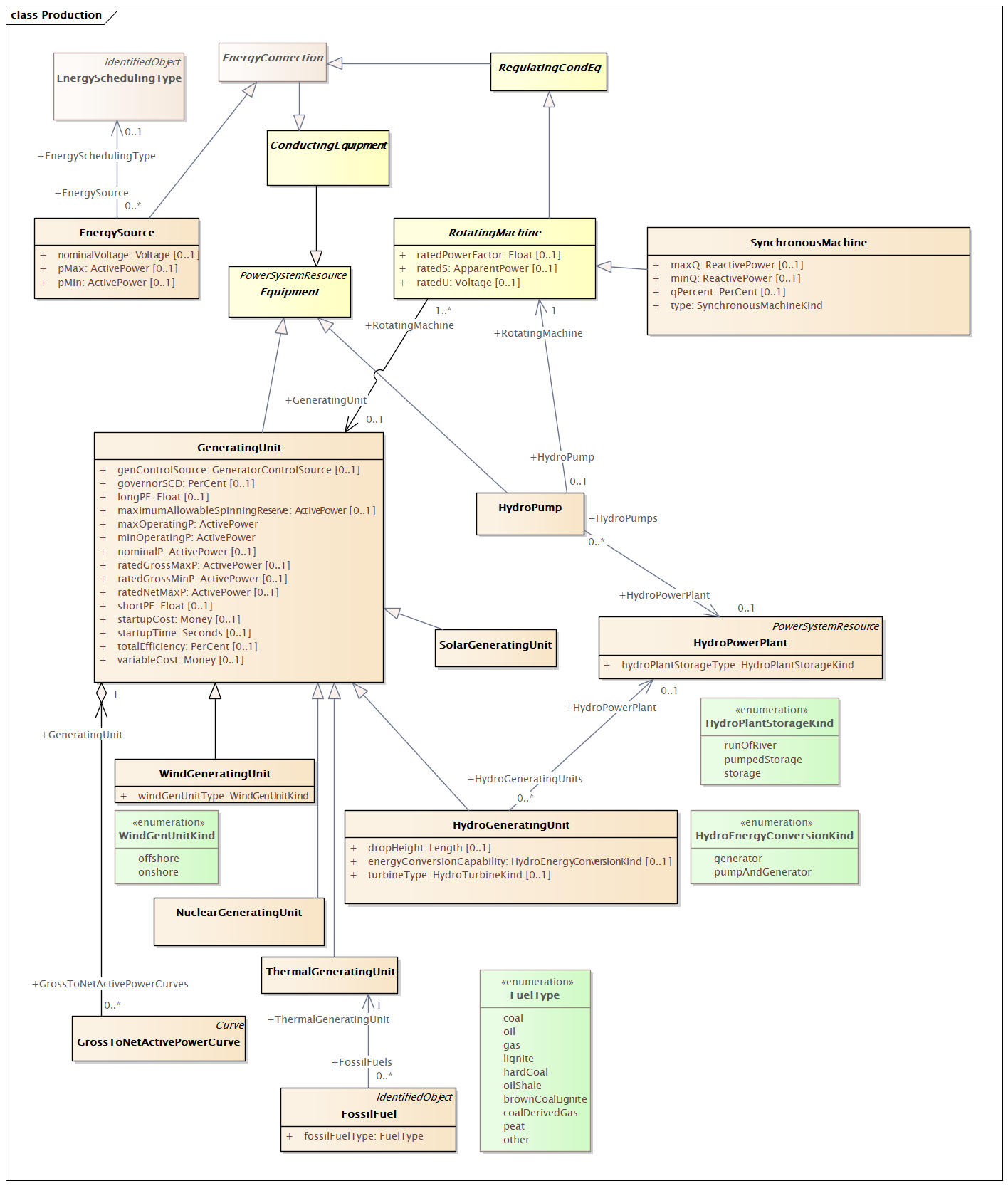


Figure 14 – Class diagram LTDSEquipmentProfile::Production

Figure 14: The diagram shows production related classes.

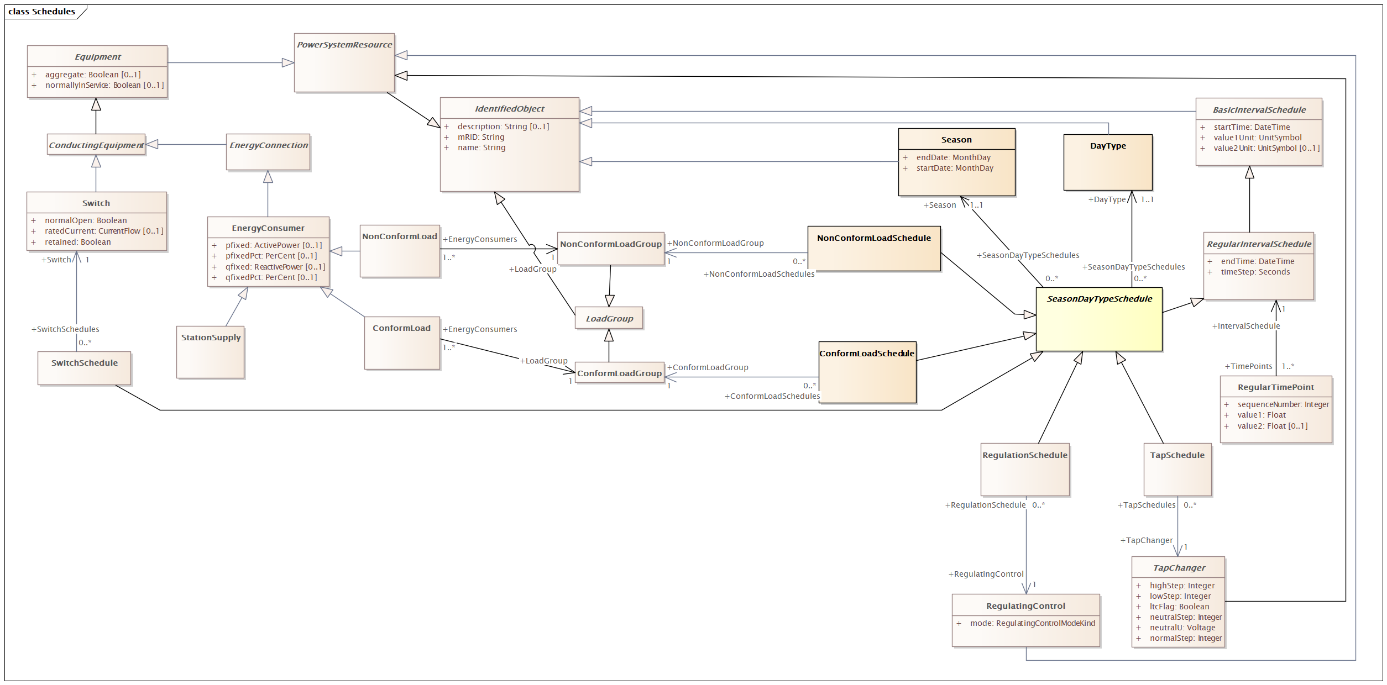


Figure 15 – Class diagram LTDSEquipmentProfile::Schedules

Figure 15: The diagram shows main classes related to schedules.

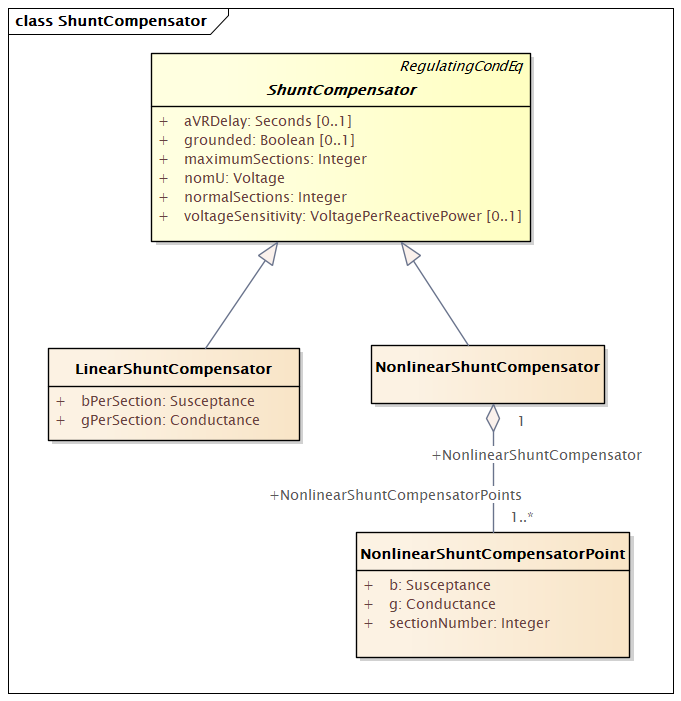


Figure 16 – Class diagram LTDSEquipmentProfile::ShuntCompensator

Figure 16: Shows the shunt compensator inheritance structure.

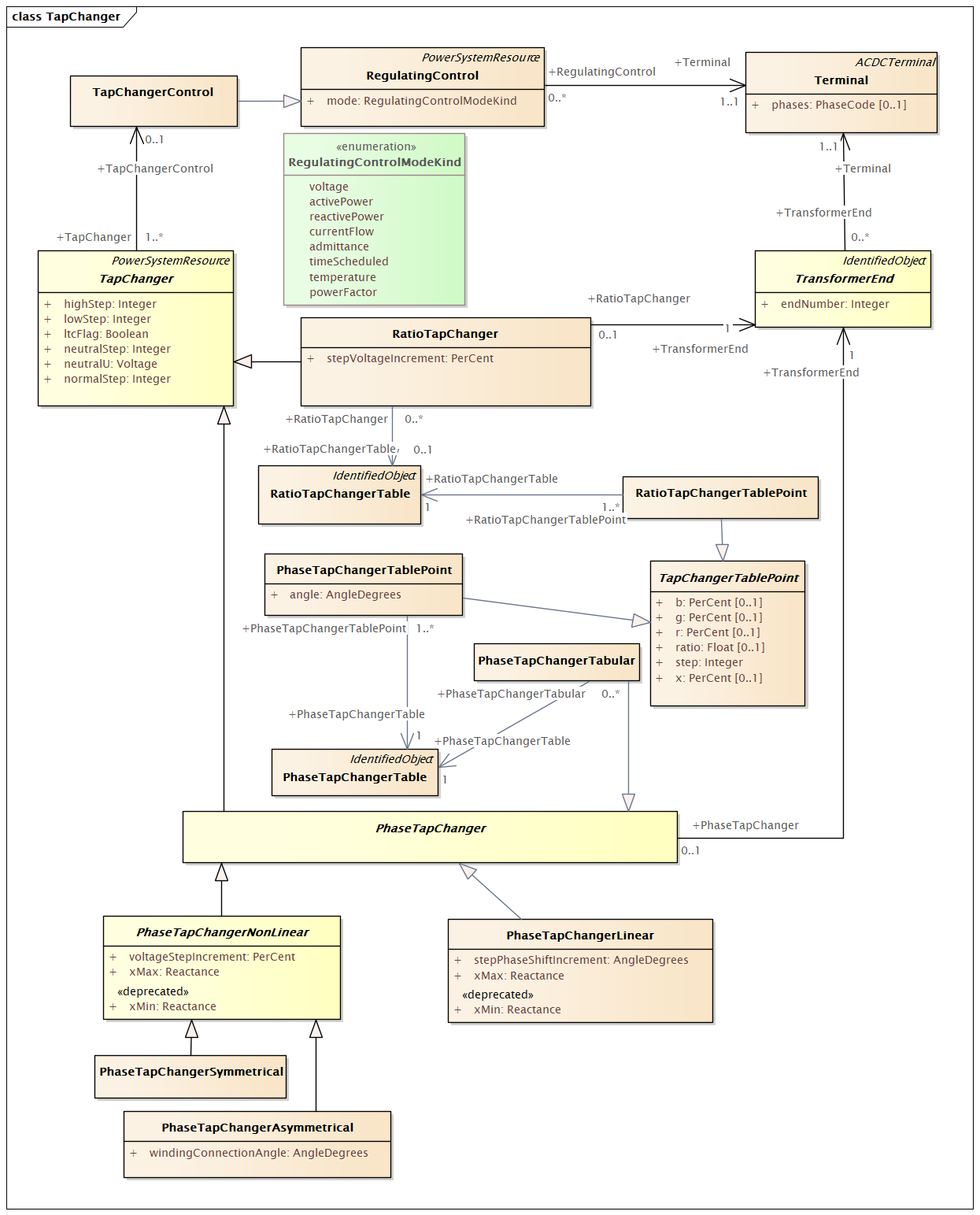


Figure 17 – Class diagram LTDSEquipmentProfile::TapChanger

Figure 17: This diagram shows all classes related to the transformer tap model.

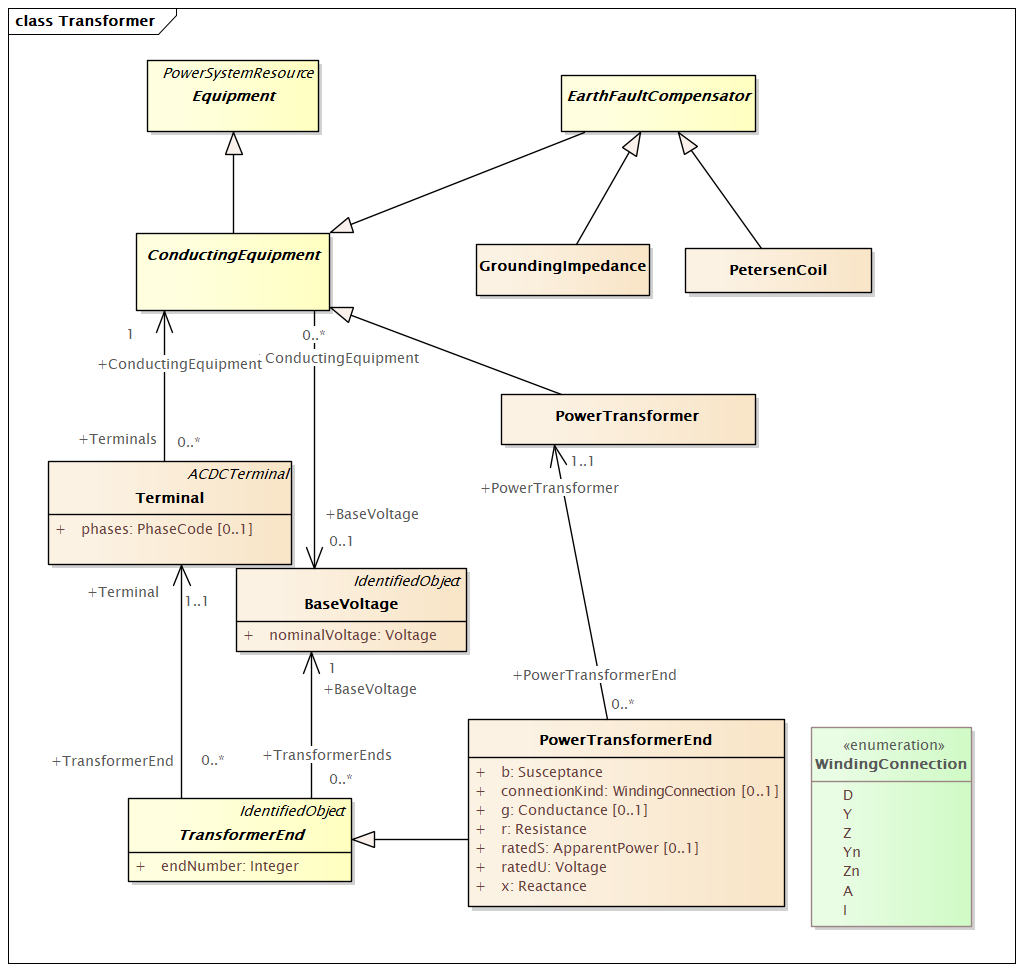


Figure 18 – Class diagram LTDSEquipmentProfile::Transformer

Figure 18: This diagram shows classes related to the core transformer model, but does not show the tap changer details.

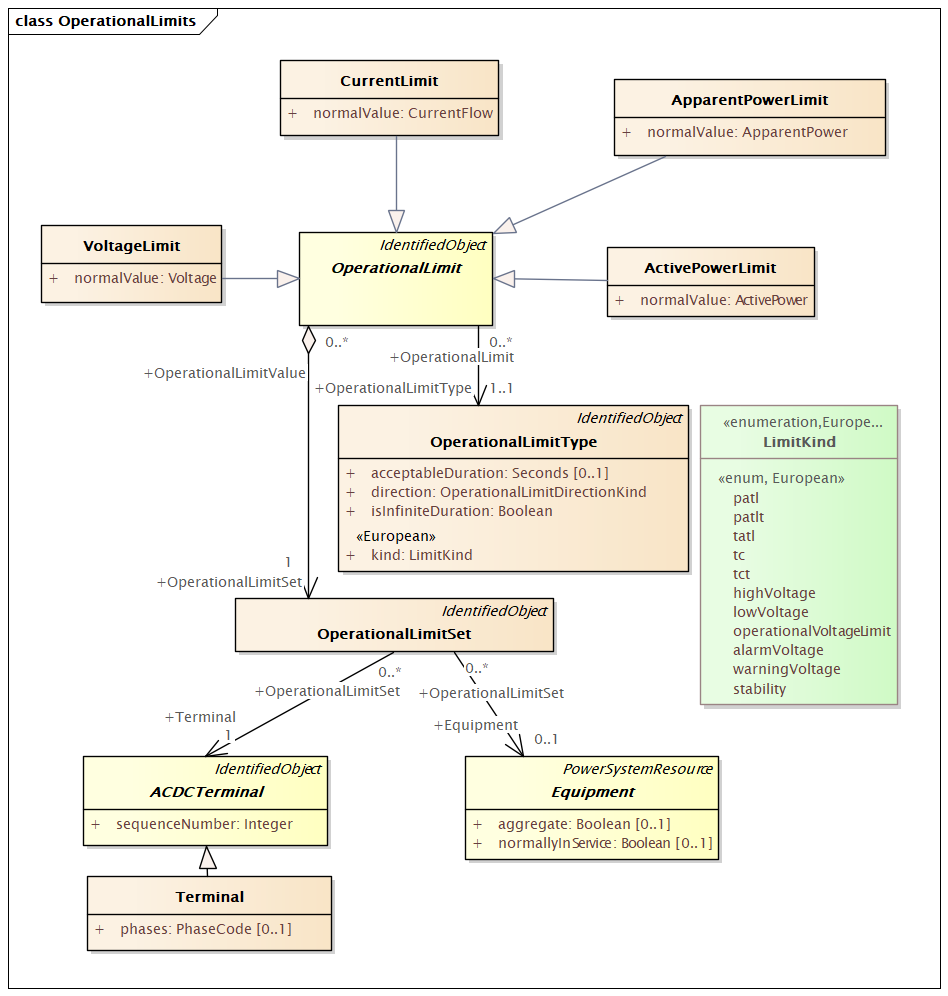


Figure 19 – Class diagram LTDSEquipmentProfile::OperationalLimits

Figure 19: This diagram shows operational limits as they tie back into the core model.

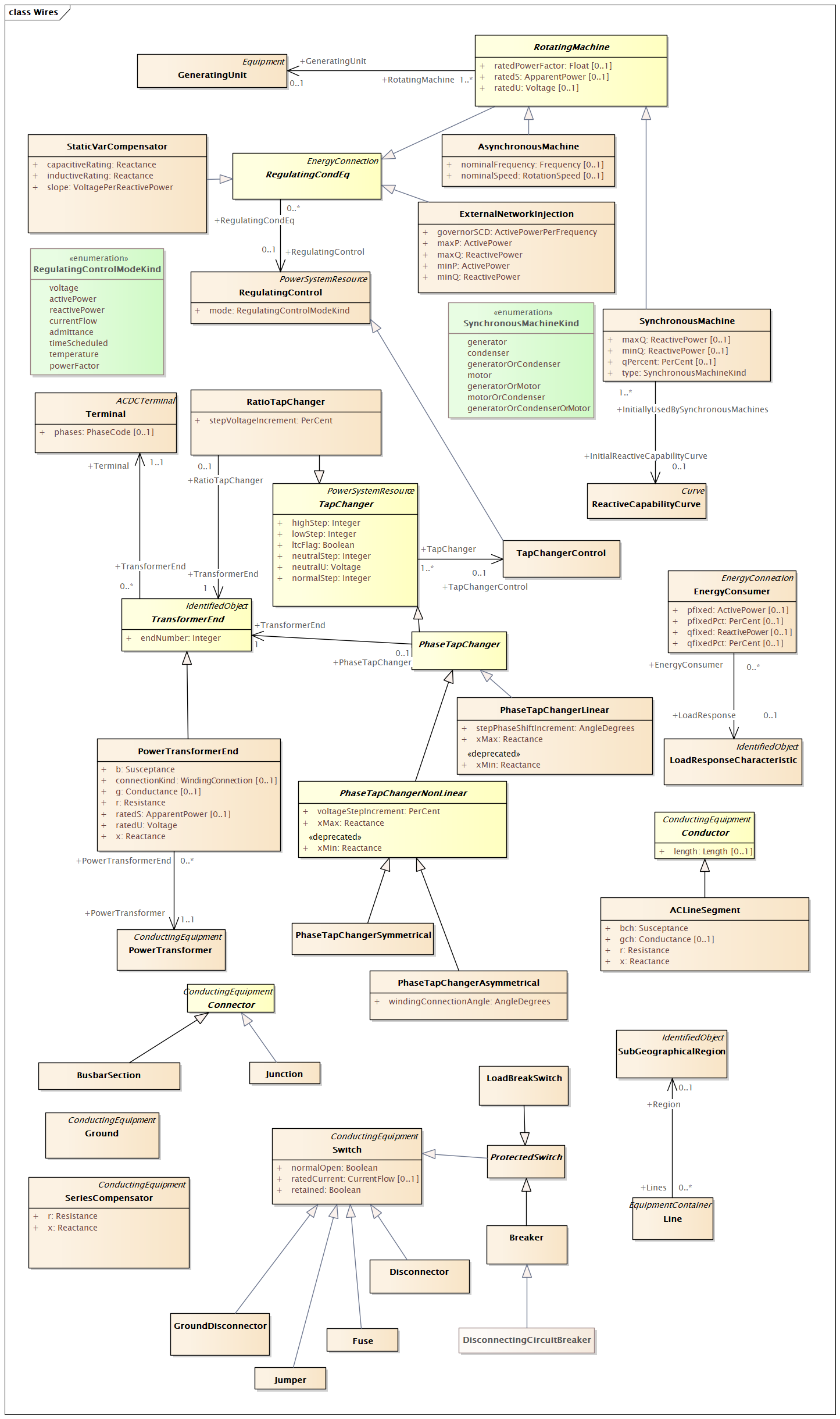


Figure 20 – Class diagram LTDSEquipmentProfile::Wires

Figure 20: The diagram shows the main classes from the wires package in the canonical CIM.

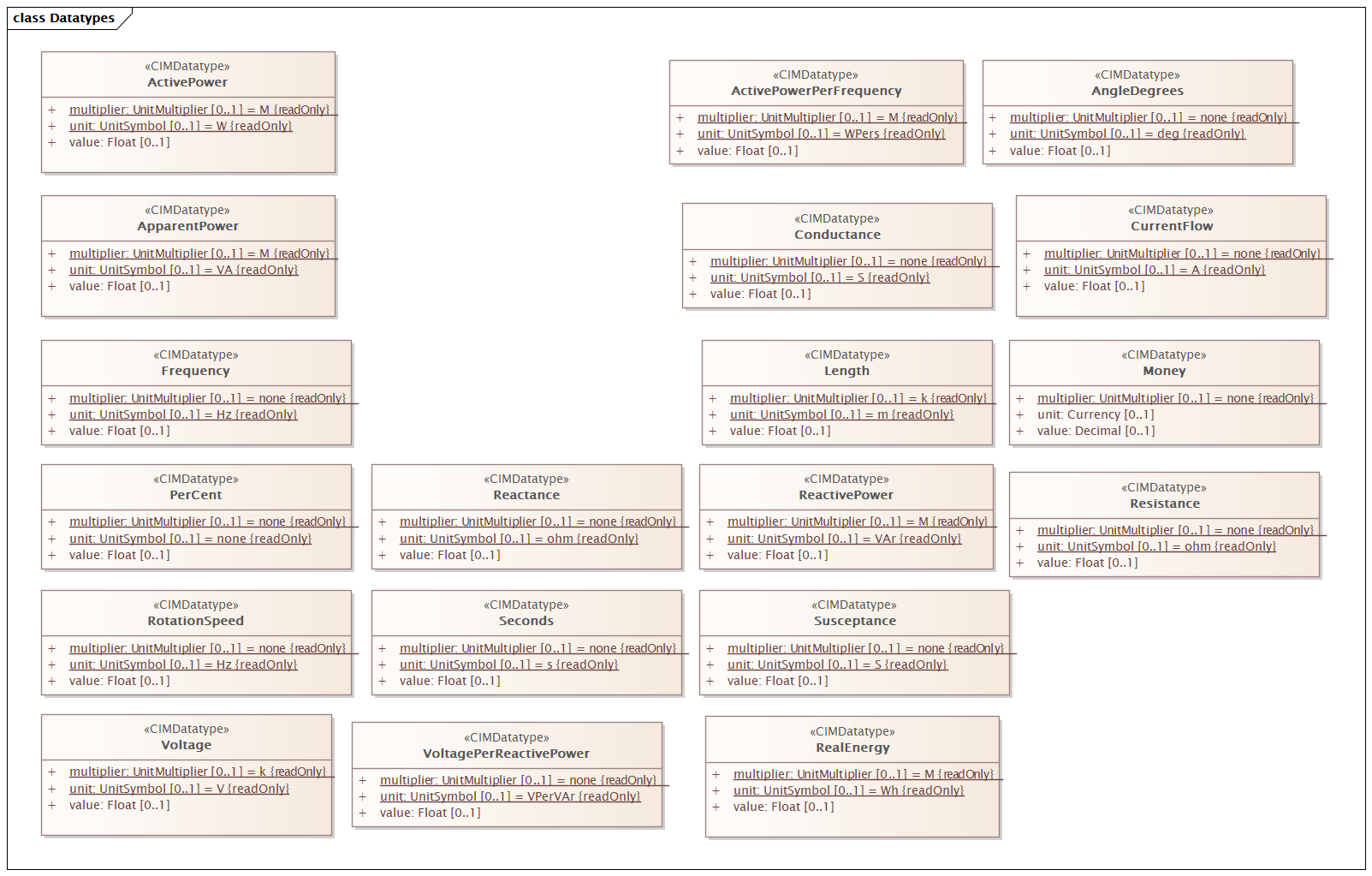


Figure 21 – Class diagram LTDSEquipmentProfile::Datatypes

Figure 21: The diagram shows datatypes that are used by classes in the profile. Stereotypes are used to describe the datatypes. The following stereotypes are defined:

<<enumeration>> A list of permissible constant values.

<<Primitive>> The most basic data types used to compose all other data types.

<<CIMDatatype>> A datatype that contains a value attribute, an optional unit of measure and a unit multiplier. The unit and multiplier may be specified as a static variable initialized to the allowed value.

<<Compound>> A composite of Primitive, enumeration, CIMDatatype or other Compound classes, as long as the Compound classes do not recurse.

For all datatypes both positive and negative values are allowed unless stated otherwise for a particular datatype.

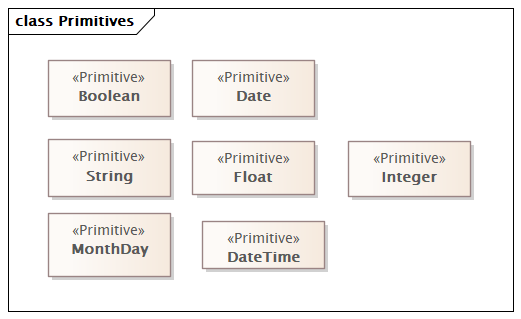


Figure 22 – Class diagram LTDSEquipmentProfile::Primitives

Figure 22: The diagram shows datatypes that are used by classes in the profile.

For all datatypes both positive and negative values are allowed unless stated otherwise for a particular datatype.

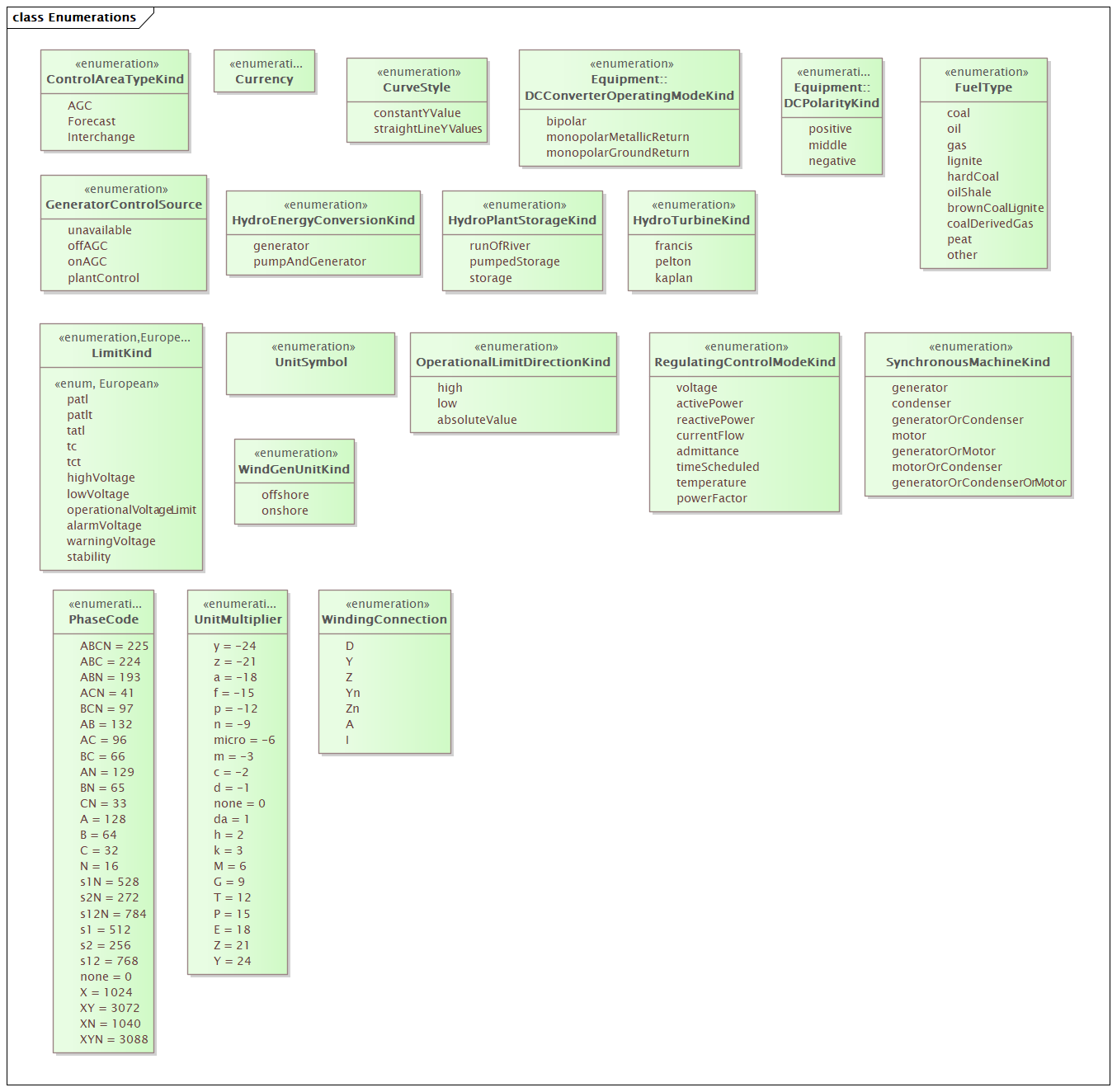


Figure 23 – Class diagram LTDSEquipmentProfile::Enumerations

Figure 23: The diagram shows enumerations that are used by classes in the profile.

### (abstract) ACDCTerminal

Inheritance path = [IdentifiedObject](#UML12)

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

Table 1 shows all attributes of ACDCTerminal.

Table 1 – Attributes of LTDSEquipmentProfile::ACDCTerminal

| name | mult | type | description |
| --- | --- | --- | --- |
| sequenceNumber | 1..1 | [Integer](#UML65) | 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](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 2 shows all association ends of ACDCTerminal with other classes.

Table 2 – Association ends of LTDSEquipmentProfile::ACDCTerminal with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 1..\* | BusNameMarker | 0..1 | [BusNameMarker](#UML1864) | The bus name marker used to name the bus (topological node). |

### ACLineSegment

Inheritance path = [Conductor](#UML1912) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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 three phase 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.

Table 3 shows all attributes of ACLineSegment.

Table 3 – Attributes of LTDSEquipmentProfile::ACLineSegment

| name | mult | type | description |
| --- | --- | --- | --- |
| bch | 1..1 | [Susceptance](#UML53) | 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. |
| gch | 0..1 | [Conductance](#UML41) | Positive sequence shunt (charging) conductance, uniformly distributed, of the entire line section. |
| r | 1..1 | [Resistance](#UML50) | Positive sequence series resistance of the entire line section. |
| x | 1..1 | [Reactance](#UML47) | Positive sequence series reactance of the entire line section. |
| length | 0..1 | [Length](#UML44) | inherited from: [Conductor](#UML1912) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 4 – Association ends of LTDSEquipmentProfile::ACLineSegment with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | PerLengthImpedance | 0..1 | [PerLengthImpedance](#UML2015) | Per-length impedance of this line segment. |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### ActivePowerLimit

Inheritance path = [OperationalLimit](#UML1894) : [IdentifiedObject](#UML12)

Limit on active power flow.

Table 5 shows all attributes of ActivePowerLimit.

Table 5 – Attributes of LTDSEquipmentProfile::ActivePowerLimit

| name | mult | type | description |
| --- | --- | --- | --- |
| normalValue | 1..1 | [ActivePower](#UML33) | The normal value of active power limit. The attribute shall be a positive value or zero. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 6 shows all association ends of ActivePowerLimit with other classes.

Table 6 – Association ends of LTDSEquipmentProfile::ActivePowerLimit with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | OperationalLimitType | 1..1 | [OperationalLimitType](#UML1900) | inherited from: [OperationalLimit](#UML1894) |
| 0..\* | OperationalLimitSet | 1..1 | [OperationalLimitSet](#UML1899) | inherited from: [OperationalLimit](#UML1894) |

### ApparentPowerLimit

Inheritance path = [OperationalLimit](#UML1894) : [IdentifiedObject](#UML12)

Apparent power limit.

Table 7 shows all attributes of ApparentPowerLimit.

Table 7 – Attributes of LTDSEquipmentProfile::ApparentPowerLimit

| name | mult | type | description |
| --- | --- | --- | --- |
| normalValue | 1..1 | [ApparentPower](#UML40) | The normal apparent power limit. The attribute shall be a positive value or zero. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 8 – Association ends of LTDSEquipmentProfile::ApparentPowerLimit with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | OperationalLimitType | 1..1 | [OperationalLimitType](#UML1900) | inherited from: [OperationalLimit](#UML1894) |
| 0..\* | OperationalLimitSet | 1..1 | [OperationalLimitSet](#UML1899) | inherited from: [OperationalLimit](#UML1894) |

### AsynchronousMachine

Inheritance path = [RotatingMachine](#UML1929) : [RegulatingCondEq](#UML1926) : [EnergyConnection](#UML1920) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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.

Table 9 shows all attributes of AsynchronousMachine.

Table 9 – Attributes of LTDSEquipmentProfile::AsynchronousMachine

| name | mult | type | description |
| --- | --- | --- | --- |
| nominalFrequency | 0..1 | [Frequency](#UML43) | Nameplate data indicates if the machine is 50 Hz or 60 Hz. |
| nominalSpeed | 0..1 | [RotationSpeed](#UML51) | Nameplate data. Depends on the slip and number of pole pairs. |
| ratedPowerFactor | 0..1 | [Float](#UML64) | inherited from: [RotatingMachine](#UML1929) |
| ratedS | 0..1 | [ApparentPower](#UML40) | inherited from: [RotatingMachine](#UML1929) |
| ratedU | 0..1 | [Voltage](#UML54) | inherited from: [RotatingMachine](#UML1929) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 10 shows all association ends of AsynchronousMachine with other classes.

Table 10 – Association ends of LTDSEquipmentProfile::AsynchronousMachine with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 1..\* | GeneratingUnit | 0..1 | [GeneratingUnit](#UML1971) | inherited from: [RotatingMachine](#UML1929) |
| 0..\* | RegulatingControl | 0..1 | [RegulatingControl](#UML1995) | inherited from: [RegulatingCondEq](#UML1926) |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### (abstract) AuxiliaryEquipment

Inheritance path = [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

AuxiliaryEquipment describe equipment that is not performing any primary functions but support for the equipment performing the primary function.

AuxiliaryEquipment is attached to primary equipment via an association with Terminal.

Table 11 shows all attributes of AuxiliaryEquipment.

Table 11 – Attributes of LTDSEquipmentProfile::AuxiliaryEquipment

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 12 shows all association ends of AuxiliaryEquipment with other classes.

Table 12 – Association ends of LTDSEquipmentProfile::AuxiliaryEquipment with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Terminal | 1..1 | [Terminal](#UML1888) | The Terminal at the equipment where the AuxiliaryEquipment is attached. |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### BaseVoltage

Inheritance path = [IdentifiedObject](#UML12)

Defines a system base voltage which is referenced.

Table 13 shows all attributes of BaseVoltage.

Table 13 – Attributes of LTDSEquipmentProfile::BaseVoltage

| name | mult | type | description |
| --- | --- | --- | --- |
| nominalVoltage | 1..1 | [Voltage](#UML54) | The power system resource's base voltage. Shall be a positive value and not zero. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### (abstract) BasicIntervalSchedule

Inheritance path = [IdentifiedObject](#UML12)

Schedule of values at points in time.

Table 14 shows all attributes of BasicIntervalSchedule.

Table 14 – Attributes of LTDSEquipmentProfile::BasicIntervalSchedule

| name | mult | type | description |
| --- | --- | --- | --- |
| startTime | 1..1 | [DateTime](#UML63) | The time for the first time point. The value can be a time of day, not a specific date. |
| value1Unit | 1..1 | [UnitSymbol](#UML30) | Value1 units of measure. |
| value2Unit | 0..1 | [UnitSymbol](#UML30) | Value2 units of measure. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### BatteryUnit

Inheritance path = [PowerElectronicsUnit](#UML1978) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

An electrochemical energy storage device.

Table 15 shows all attributes of BatteryUnit.

Table 15 – Attributes of LTDSEquipmentProfile::BatteryUnit

| name | mult | type | description |
| --- | --- | --- | --- |
| ratedE | 1..1 | [RealEnergy](#UML49) | Full energy storage capacity of the battery. The attribute shall be a positive value. |
| maxP | 0..1 | [ActivePower](#UML33) | inherited from: [PowerElectronicsUnit](#UML1978) |
| minP | 0..1 | [ActivePower](#UML33) | inherited from: [PowerElectronicsUnit](#UML1978) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 16 shows all association ends of BatteryUnit with other classes.

Table 16 – Association ends of LTDSEquipmentProfile::BatteryUnit with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### Bay

Inheritance path = [EquipmentContainer](#UML1985) : [ConnectivityNodeContainer](#UML1984) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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.

Table 17 shows all attributes of Bay.

Table 17 – Attributes of LTDSEquipmentProfile::Bay

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 18 shows all association ends of Bay with other classes.

Table 18 – Association ends of LTDSEquipmentProfile::Bay with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | VoltageLevel | 1..1 | [VoltageLevel](#UML1992) | The voltage level containing this bay. |

### (European) BoundaryPoint

Inheritance path = [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

Designates a connection point at which one or more model authority sets shall connect to. The location of the connection point as well as other properties are agreed between organisations responsible for the interconnection, hence all attributes of the class represent this agreement. It is primarily used in a boundary model authority set which can contain one or many BoundaryPoint-s among other Equipment-s and their connections.

Table 19 shows all attributes of BoundaryPoint.

Table 19 – Attributes of LTDSEquipmentProfile::BoundaryPoint

| name | mult | type | description |
| --- | --- | --- | --- |
| fromEndIsoCode | 1..1 | [String](#UML67) | (European) The ISO code of the region which the "From" side of the Boundary point belongs to or it is connected to.  The ISO code is a two-character country code as defined by ISO 3166 (http://www.iso.org/iso/country\_codes). The length of the string is 2 characters maximum. |
| fromEndName | 1..1 | [String](#UML67) | (European) A human readable name with length of the string 64 characters maximum. It covers the following two cases:  -if the Boundary point is placed on a tie-line, it is the name (IdentifiedObject.name) of the substation at which the "From" side of the tie-line is connected to.  -if the Boundary point is placed in a substation, it is the name (IdentifiedObject.name) of the element (e.g. PowerTransformer, ACLineSegment, Switch, etc.) at which the "From" side of the Boundary point is connected to. |
| fromEndNameTso | 1..1 | [String](#UML67) | (European) Identifies the name of the transmission system operator, distribution system operator or other entity at which the "From" side of the interconnection is connected to. The length of the string is 64 characters maximum. |
| toEndIsoCode | 1..1 | [String](#UML67) | (European) The ISO code of the region which the "To" side of the Boundary point belongs to or is connected to.  The ISO code is a two-character country code as defined by ISO 3166 (http://www.iso.org/iso/country\_codes). The length of the string is 2 characters maximum. |
| toEndName | 1..1 | [String](#UML67) | (European) A human readable name with length of the string 64 characters maximum. It covers the following two cases:  -if the Boundary point is placed on a tie-line, it is the name (IdentifiedObject.name) of the substation at which the "To" side of the tie-line is connected to.  -if the Boundary point is placed in a substation, it is the name (IdentifiedObject.name) of the element (e.g. PowerTransformer, ACLineSegment, Switch, etc.) at which the "To" side of the Boundary point is connected to. |
| toEndNameTso | 1..1 | [String](#UML67) | (European) Identifies the name of the transmission system operator, distribution system operator or other entity at which the "To" side of the interconnection is connected to. The length of the string is 64 characters maximum. |
| isDirectCurrent | 0..1 | [Boolean](#UML57) | (European) If true, this boundary point is a point of common coupling (PCC) of a direct current (DC) interconnection, otherwise the interconnection is AC (default). |
| isExcludedFromAreaInterchange | 0..1 | [Boolean](#UML57) | (European) If true, this boundary point is on the interconnection that is excluded from control area interchange calculation and consequently has no related tie flows. Otherwise, the interconnection is included in control area interchange and a TieFlow is required at all sides of the boundary point (default). |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 20 – Association ends of LTDSEquipmentProfile::BoundaryPoint with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..1 | ConnectivityNode | 1..1 | [ConnectivityNode](#UML1874) | (European) The connectivity node that is designated as a boundary point. |

### Breaker

Inheritance path = [ProtectedSwitch](#UML1944) : [Switch](#UML1943) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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.

Table 21 shows all attributes of Breaker.

Table 21 – Attributes of LTDSEquipmentProfile::Breaker

| name | mult | type | description |
| --- | --- | --- | --- |
| normalOpen | 1..1 | [Boolean](#UML57) | inherited from: [Switch](#UML1943) |
| ratedCurrent | 0..1 | [CurrentFlow](#UML42) | inherited from: [Switch](#UML1943) |
| retained | 1..1 | [Boolean](#UML57) | inherited from: [Switch](#UML1943) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 22 – Association ends of LTDSEquipmentProfile::Breaker with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### BusbarSection

Inheritance path = [Connector](#UML1914) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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 voltage transformers 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.

Table 23 shows all attributes of BusbarSection.

Table 23 – Attributes of LTDSEquipmentProfile::BusbarSection

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 24 – Association ends of LTDSEquipmentProfile::BusbarSection with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### BusNameMarker

Inheritance path = [IdentifiedObject](#UML12)

Used to apply user standard names to TopologicalNodes. 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 BusbarSection terminals in the ring are typically associated. For a breaker and a half scheme, both BusbarSections would normally be associated. For a ring bus, all BusbarSections would normally be associated. For a "straight" busbar configuration, normally only the main terminal at the BusbarSection would be associated.

Table 25 shows all attributes of BusNameMarker.

Table 25 – Attributes of LTDSEquipmentProfile::BusNameMarker

| name | mult | type | description |
| --- | --- | --- | --- |
| priority | 0..1 | [Integer](#UML65) | Priority of bus name marker for use as topology bus name. Use 0 for do not care. Use 1 for highest priority. Use 2 as priority is less than 1 and so on. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 26 shows all association ends of BusNameMarker with other classes.

Table 26 – Association ends of LTDSEquipmentProfile::BusNameMarker with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | ReportingGroup | 0..1 | [ReportingGroup](#UML2008) | The reporting group to which this bus name marker belongs. |

### CAESPlant

Inheritance path = [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

Compressed air energy storage plant.

Table 27 shows all attributes of CAESPlant.

Table 27 – Attributes of LTDSEquipmentProfile::CAESPlant

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### Clamp

Inheritance path = [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A Clamp is a galvanic connection at a line segment where other equipment is connected. A Clamp does not cut the line segment.

A Clamp is ConductingEquipment and has one Terminal with an associated ConnectivityNode. Any other ConductingEquipment can be connected to the Clamp ConnectivityNode.

Table 28 shows all attributes of Clamp.

Table 28 – Attributes of LTDSEquipmentProfile::Clamp

| name | mult | type | description |
| --- | --- | --- | --- |
| lengthFromTerminal1 | 0..1 | [Length](#UML44) | The length to the place where the clamp is located starting from side one of the line segment, i.e. the line segment terminal with sequence number equal to 1. |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 29 shows all association ends of Clamp with other classes.

Table 29 – Association ends of LTDSEquipmentProfile::Clamp with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | ACLineSegment | 1..1 | [ACLineSegment](#UML1913) | The line segment to which the clamp is connected. |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### ConformLoadSchedule

Inheritance path = [SeasonDayTypeSchedule](#UML1877) : [RegularIntervalSchedule](#UML1876) : [BasicIntervalSchedule](#UML1875) : [IdentifiedObject](#UML12)

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.

Table 30 shows all attributes of ConformLoadSchedule.

Table 30 – Attributes of LTDSEquipmentProfile::ConformLoadSchedule

| name | mult | type | description |
| --- | --- | --- | --- |
| timeStep | 1..1 | [Seconds](#UML52) | inherited from: [RegularIntervalSchedule](#UML1876) |
| endTime | 1..1 | [DateTime](#UML63) | inherited from: [RegularIntervalSchedule](#UML1876) |
| startTime | 1..1 | [DateTime](#UML63) | inherited from: [BasicIntervalSchedule](#UML1875) |
| value1Unit | 1..1 | [UnitSymbol](#UML30) | inherited from: [BasicIntervalSchedule](#UML1875) |
| value2Unit | 0..1 | [UnitSymbol](#UML30) | inherited from: [BasicIntervalSchedule](#UML1875) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 31 shows all association ends of ConformLoadSchedule with other classes.

Table 31 – Association ends of LTDSEquipmentProfile::ConformLoadSchedule with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | ConformLoadGroup | 1..1 | [ConformLoadGroup](#UML1891) | The ConformLoadGroup where the ConformLoadSchedule belongs. |
| 0..\* | DayType | 1..1 | [DayType](#UML1868) | inherited from: [SeasonDayTypeSchedule](#UML1877) |
| 0..\* | Season | 1..1 | [Season](#UML2009) | inherited from: [SeasonDayTypeSchedule](#UML1877) |

### CogenerationPlant

Inheritance path = [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A set of thermal generating units for the production of electrical energy and process steam (usually from the output of the steam turbines). The steam sendout is typically used for industrial purposes or for municipal heating and cooling.

Table 32 shows all attributes of CogenerationPlant.

Table 32 – Attributes of LTDSEquipmentProfile::CogenerationPlant

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### CombinedCyclePlant

Inheritance path = [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A set of combustion turbines and steam turbines where the exhaust heat from the combustion turbines is recovered to make steam for the steam turbines, resulting in greater overall plant efficiency.

Table 33 shows all attributes of CombinedCyclePlant.

Table 33 – Attributes of LTDSEquipmentProfile::CombinedCyclePlant

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### (abstract) ConductingEquipment

Inheritance path = [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 34 shows all attributes of ConductingEquipment.

Table 34 – Attributes of LTDSEquipmentProfile::ConductingEquipment

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 35 – Association ends of LTDSEquipmentProfile::ConductingEquipment with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | Base voltage of this conducting equipment. Use only when there is no voltage level container used and only one base voltage applies. For example, not used for transformers. |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### (abstract) Conductor

Inheritance path = [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 36 shows all attributes of Conductor.

Table 36 – Attributes of LTDSEquipmentProfile::Conductor

| name | mult | type | description |
| --- | --- | --- | --- |
| length | 0..1 | [Length](#UML44) | Segment length for calculating line section capabilities. |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 37 – Association ends of LTDSEquipmentProfile::Conductor with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### ConformLoad

Inheritance path = [EnergyConsumer](#UML1921) : [EnergyConnection](#UML1920) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

ConformLoad represent loads that follow a daily load change pattern where the pattern can be used to scale the load with a system load.

Table 38 shows all attributes of ConformLoad.

Table 38 – Attributes of LTDSEquipmentProfile::ConformLoad

| name | mult | type | description |
| --- | --- | --- | --- |
| pfixed | 0..1 | [ActivePower](#UML33) | inherited from: [EnergyConsumer](#UML1921) |
| pfixedPct | 0..1 | [PerCent](#UML46) | inherited from: [EnergyConsumer](#UML1921) |
| qfixed | 0..1 | [ReactivePower](#UML48) | inherited from: [EnergyConsumer](#UML1921) |
| qfixedPct | 0..1 | [PerCent](#UML46) | inherited from: [EnergyConsumer](#UML1921) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 39 – Association ends of LTDSEquipmentProfile::ConformLoad with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 1..\* | LoadGroup | 1..1 | [ConformLoadGroup](#UML1891) | Group of this ConformLoad. |
| 0..\* | LoadResponse | 0..1 | [LoadResponseCharacteristic](#UML1893) | inherited from: [EnergyConsumer](#UML1921) |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### ConformLoadGroup

Inheritance path = [LoadGroup](#UML1890) : [IdentifiedObject](#UML12)

A group of loads conforming to an allocation pattern.

Table 40 shows all attributes of ConformLoadGroup.

Table 40 – Attributes of LTDSEquipmentProfile::ConformLoadGroup

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 41 – Association ends of LTDSEquipmentProfile::ConformLoadGroup with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 1..\* | SubLoadArea | 1..1 | [SubLoadArea](#UML1867) | inherited from: [LoadGroup](#UML1890) |

### ConnectivityNode

Inheritance path = [IdentifiedObject](#UML12)

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

Table 42 shows all attributes of ConnectivityNode.

Table 42 – Attributes of LTDSEquipmentProfile::ConnectivityNode

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 43 – Association ends of LTDSEquipmentProfile::ConnectivityNode with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | ConnectivityNodeContainer | 1..1 | [ConnectivityNodeContainer](#UML1984) | Container of this connectivity node. |

### (abstract) ConnectivityNodeContainer

Inheritance path = [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 44 shows all attributes of ConnectivityNodeContainer.

Table 44 – Attributes of LTDSEquipmentProfile::ConnectivityNodeContainer

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### (abstract) Connector

Inheritance path = [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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.

Table 45 shows all attributes of Connector.

Table 45 – Attributes of LTDSEquipmentProfile::Connector

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 46 shows all association ends of Connector with other classes.

Table 46 – Association ends of LTDSEquipmentProfile::Connector with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### ControlArea

Inheritance path = [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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, power flow solution area interchange control specification, and input to load forecasting. All generation and load within the area defined by the terminals on the border are considered in the area interchange control. Note that any number of overlapping control area specifications can be superimposed on the physical model. The following general principles apply to ControlArea:

1. The control area orientation for net interchange is positive for an import, negative for an export.

2. The control area net interchange is determined by summing flows in Terminals. The Terminals are identified by creating a set of TieFlow objects associated with a ControlArea object. Each TieFlow object identifies one Terminal.

3. In a single network model, a tie between two control areas must be modelled in both control area specifications, such that the two representations of the tie flow sum to zero.

4. The normal orientation of Terminal flow is positive for flow into the conducting equipment that owns the Terminal. (i.e. flow from a bus into a device is positive.) However, the orientation of each flow in the control area specification must align with the control area convention, i.e. import is positive. If the orientation of the Terminal flow referenced by a TieFlow is positive into the control area, then this is confirmed by setting TieFlow.positiveFlowIn flag TRUE. If not, the orientation must be reversed by setting the TieFlow.positiveFlowIn flag FALSE.

Table 47 shows all attributes of ControlArea.

Table 47 – Attributes of LTDSEquipmentProfile::ControlArea

| name | mult | type | description |
| --- | --- | --- | --- |
| type | 1..1 | [ControlAreaTypeKind](#UML16) | 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](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 48 shows all association ends of ControlArea with other classes.

Table 48 – Association ends of LTDSEquipmentProfile::ControlArea with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..1 | EnergyArea | 1..1 | [EnergyArea](#UML1865) | The energy area that is forecast from this control area specification. |

### ControlAreaGeneratingUnit

Inheritance path = [IdentifiedObject](#UML12)

A control area generating unit. This class is needed so that alternate control area definitions may include the same generating unit. It should be noted that only one instance within a control area should reference a specific generating unit.

Table 49 shows all attributes of ControlAreaGeneratingUnit.

Table 49 – Attributes of LTDSEquipmentProfile::ControlAreaGeneratingUnit

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 50 shows all association ends of ControlAreaGeneratingUnit with other classes.

Table 50 – Association ends of LTDSEquipmentProfile::ControlAreaGeneratingUnit with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | ControlArea | 1..1 | [ControlArea](#UML1983) | The parent control area for the generating unit specifications. |
| 0..\* | GeneratingUnit | 1..1 | [GeneratingUnit](#UML1971) | The generating unit specified for this control area. Note that a control area should include a GeneratingUnit only once. |

### CurrentLimit

Inheritance path = [OperationalLimit](#UML1894) : [IdentifiedObject](#UML12)

Operational limit on current.

Table 51 shows all attributes of CurrentLimit.

Table 51 – Attributes of LTDSEquipmentProfile::CurrentLimit

| name | mult | type | description |
| --- | --- | --- | --- |
| normalValue | 1..1 | [CurrentFlow](#UML42) | The normal value for limit on current flow. The attribute shall be a positive value or zero. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 52 shows all association ends of CurrentLimit with other classes.

Table 52 – Association ends of LTDSEquipmentProfile::CurrentLimit with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | OperationalLimitType | 1..1 | [OperationalLimitType](#UML1900) | inherited from: [OperationalLimit](#UML1894) |
| 0..\* | OperationalLimitSet | 1..1 | [OperationalLimitSet](#UML1899) | inherited from: [OperationalLimit](#UML1894) |

### CurrentTransformer

Inheritance path = [Sensor](#UML1955) : [AuxiliaryEquipment](#UML1953) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

Instrument transformer used to measure electrical qualities of the circuit that is being protected and/or monitored. Typically used as current transducer for the purpose of metering or protection. A typical secondary current rating would be 5A.

Table 53 shows all attributes of CurrentTransformer.

Table 53 – Attributes of LTDSEquipmentProfile::CurrentTransformer

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 54 – Association ends of LTDSEquipmentProfile::CurrentTransformer with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Terminal | 1..1 | [Terminal](#UML1888) | inherited from: [AuxiliaryEquipment](#UML1953) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### (abstract) Curve

Inheritance path = [IdentifiedObject](#UML12)

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

Table 55 shows all attributes of Curve.

Table 55 – Attributes of LTDSEquipmentProfile::Curve

| name | mult | type | description |
| --- | --- | --- | --- |
| curveStyle | 1..1 | [CurveStyle](#UML18) | The style or shape of the curve. |
| xUnit | 1..1 | [UnitSymbol](#UML30) | The X-axis units of measure. |
| y1Unit | 1..1 | [UnitSymbol](#UML30) | The Y1-axis units of measure. |
| y2Unit | 0..1 | [UnitSymbol](#UML30) | The Y2-axis units of measure. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### CurveData root class

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.

Table 56 shows all attributes of CurveData.

Table 56 – Attributes of LTDSEquipmentProfile::CurveData

| name | mult | type | description |
| --- | --- | --- | --- |
| xvalue | 1..1 | [Float](#UML64) | The data value of the X-axis variable, depending on the X-axis units. |
| y1value | 1..1 | [Float](#UML64) | The data value of the first Y-axis variable, depending on the Y-axis units. |
| y2value | 0..1 | [Float](#UML64) | The data value of the second Y-axis variable (if present), depending on the Y-axis units. |

Table 57 shows all association ends of CurveData with other classes.

Table 57 – Association ends of LTDSEquipmentProfile::CurveData with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 1..\* | Curve | 1..1 | [Curve](#UML1869) | The curve of this curve data point. |

### Cut

Inheritance path = [Switch](#UML1943) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A cut separates a line segment into two parts. The cut appears as a switch inserted between these two parts and connects them together. As the cut is normally open there is no galvanic connection between the two line segment parts. But it is possible to close the cut to get galvanic connection.

The cut terminals are oriented towards the line segment terminals with the same sequence number. Hence the cut terminal with sequence number equal to 1 is oriented to the line segment's terminal with sequence number equal to 1.

The cut terminals also act as connection points for jumpers and other equipment, e.g. a mobile generator. To enable this, connectivity nodes are placed at the cut terminals. Once the connectivity nodes are in place any conducting equipment can be connected at them.

Table 58 shows all attributes of Cut.

Table 58 – Attributes of LTDSEquipmentProfile::Cut

| name | mult | type | description |
| --- | --- | --- | --- |
| lengthFromTerminal1 | 0..1 | [Length](#UML44) | The length to the place where the cut is located starting from side one of the cut line segment, i.e. the line segment Terminal with sequenceNumber equal to 1. |
| normalOpen | 1..1 | [Boolean](#UML57) | inherited from: [Switch](#UML1943) |
| ratedCurrent | 0..1 | [CurrentFlow](#UML42) | inherited from: [Switch](#UML1943) |
| retained | 1..1 | [Boolean](#UML57) | inherited from: [Switch](#UML1943) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 59 shows all association ends of Cut with other classes.

Table 59 – Association ends of LTDSEquipmentProfile::Cut with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | ACLineSegment | 1..1 | [ACLineSegment](#UML1913) | The line segment to which the cut is applied. |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### DayType

Inheritance path = [IdentifiedObject](#UML12)

Group of similar days. For example it could be used to represent weekdays, weekend, or holidays.

Table 60 shows all attributes of DayType.

Table 60 – Attributes of LTDSEquipmentProfile::DayType

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### RegulationSchedule

Inheritance path = [SeasonDayTypeSchedule](#UML1877) : [RegularIntervalSchedule](#UML1876) : [BasicIntervalSchedule](#UML1875) : [IdentifiedObject](#UML12)

A pre-established pattern over time for a controlled variable, e.g., busbar voltage.

Table 61 shows all attributes of RegulationSchedule.

Table 61 – Attributes of LTDSEquipmentProfile::RegulationSchedule

| name | mult | type | description |
| --- | --- | --- | --- |
| timeStep | 1..1 | [Seconds](#UML52) | inherited from: [RegularIntervalSchedule](#UML1876) |
| endTime | 1..1 | [DateTime](#UML63) | inherited from: [RegularIntervalSchedule](#UML1876) |
| startTime | 1..1 | [DateTime](#UML63) | inherited from: [BasicIntervalSchedule](#UML1875) |
| value1Unit | 1..1 | [UnitSymbol](#UML30) | inherited from: [BasicIntervalSchedule](#UML1875) |
| value2Unit | 0..1 | [UnitSymbol](#UML30) | inherited from: [BasicIntervalSchedule](#UML1875) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 62 – Association ends of LTDSEquipmentProfile::RegulationSchedule with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | RegulatingControl | 1..1 | [RegulatingControl](#UML1995) | Regulating controls that have this schedule. |
| 0..\* | DayType | 1..1 | [DayType](#UML1868) | inherited from: [SeasonDayTypeSchedule](#UML1877) |
| 0..\* | Season | 1..1 | [Season](#UML2009) | inherited from: [SeasonDayTypeSchedule](#UML1877) |

### Disconnector

Inheritance path = [Switch](#UML1943) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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.

Table 63 shows all attributes of Disconnector.

Table 63 – Attributes of LTDSEquipmentProfile::Disconnector

| name | mult | type | description |
| --- | --- | --- | --- |
| normalOpen | 1..1 | [Boolean](#UML57) | inherited from: [Switch](#UML1943) |
| ratedCurrent | 0..1 | [CurrentFlow](#UML42) | inherited from: [Switch](#UML1943) |
| retained | 1..1 | [Boolean](#UML57) | inherited from: [Switch](#UML1943) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 64 – Association ends of LTDSEquipmentProfile::Disconnector with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### DisconnectingCircuitBreaker

Inheritance path = [Breaker](#UML1946) : [ProtectedSwitch](#UML1944) : [Switch](#UML1943) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A circuit breaking device including disconnecting function, eliminating the need for separate disconnectors.

Table 65 shows all attributes of DisconnectingCircuitBreaker.

Table 65 – Attributes of LTDSEquipmentProfile::DisconnectingCircuitBreaker

| name | mult | type | description |
| --- | --- | --- | --- |
| normalOpen | 1..1 | [Boolean](#UML57) | inherited from: [Switch](#UML1943) |
| ratedCurrent | 0..1 | [CurrentFlow](#UML42) | inherited from: [Switch](#UML1943) |
| retained | 1..1 | [Boolean](#UML57) | inherited from: [Switch](#UML1943) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 66 – Association ends of LTDSEquipmentProfile::DisconnectingCircuitBreaker with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### (abstract) EarthFaultCompensator

Inheritance path = [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 67 shows all attributes of EarthFaultCompensator.

Table 67 – Attributes of LTDSEquipmentProfile::EarthFaultCompensator

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 68 – Association ends of LTDSEquipmentProfile::EarthFaultCompensator with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### (abstract) EnergyArea

Inheritance path = [IdentifiedObject](#UML12)

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.

Table 69 shows all attributes of EnergyArea.

Table 69 – Attributes of LTDSEquipmentProfile::EnergyArea

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### (abstract) EnergyConnection

Inheritance path = [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A connection of energy generation or consumption on the power system model.

Table 70 shows all attributes of EnergyConnection.

Table 70 – Attributes of LTDSEquipmentProfile::EnergyConnection

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 71 shows all association ends of EnergyConnection with other classes.

Table 71 – Association ends of LTDSEquipmentProfile::EnergyConnection with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### EnergyConsumer

Inheritance path = [EnergyConnection](#UML1920) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

EnergyConsumer.pfixed, .qfixed, .pfixedPct and .qfixedPct have meaning only if there is no LoadResponseCharacteristic associated with EnergyConsumer or if LoadResponseCharacteristic.exponentModel is set to False.

Table 72 shows all attributes of EnergyConsumer.

Table 72 – Attributes of LTDSEquipmentProfile::EnergyConsumer

| name | mult | type | description |
| --- | --- | --- | --- |
| pfixed | 0..1 | [ActivePower](#UML33) | Active power of the load that is a fixed quantity and does not vary as load group value varies. Load sign convention is used, i.e. positive sign means flow out from a node. |
| pfixedPct | 0..1 | [PerCent](#UML46) | Fixed active power as a percentage of load group fixed active power. Used to represent the time-varying components. Load sign convention is used, i.e. positive sign means flow out from a node. |
| qfixed | 0..1 | [ReactivePower](#UML48) | Reactive power of the load that is a fixed quantity and does not vary as load group value varies. Load sign convention is used, i.e. positive sign means flow out from a node. |
| qfixedPct | 0..1 | [PerCent](#UML46) | Fixed reactive power as a percentage of load group fixed reactive power. Used to represent the time-varying components. Load sign convention is used, i.e. positive sign means flow out from a node. |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 73 shows all association ends of EnergyConsumer with other classes.

Table 73 – Association ends of LTDSEquipmentProfile::EnergyConsumer with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | LoadResponse | 0..1 | [LoadResponseCharacteristic](#UML1893) | The load response characteristic of this load. If missing, this load is assumed to be constant power. |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### EnergySchedulingType

Inheritance path = [IdentifiedObject](#UML12)

Used to define the type of generation for scheduling purposes.

Table 74 shows all attributes of EnergySchedulingType.

Table 74 – Attributes of LTDSEquipmentProfile::EnergySchedulingType

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### TapSchedule

Inheritance path = [SeasonDayTypeSchedule](#UML1877) : [RegularIntervalSchedule](#UML1876) : [BasicIntervalSchedule](#UML1875) : [IdentifiedObject](#UML12)

A pre-established pattern over time for a tap step.

Table 75 shows all attributes of TapSchedule.

Table 75 – Attributes of LTDSEquipmentProfile::TapSchedule

| name | mult | type | description |
| --- | --- | --- | --- |
| timeStep | 1..1 | [Seconds](#UML52) | inherited from: [RegularIntervalSchedule](#UML1876) |
| endTime | 1..1 | [DateTime](#UML63) | inherited from: [RegularIntervalSchedule](#UML1876) |
| startTime | 1..1 | [DateTime](#UML63) | inherited from: [BasicIntervalSchedule](#UML1875) |
| value1Unit | 1..1 | [UnitSymbol](#UML30) | inherited from: [BasicIntervalSchedule](#UML1875) |
| value2Unit | 0..1 | [UnitSymbol](#UML30) | inherited from: [BasicIntervalSchedule](#UML1875) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 76 – Association ends of LTDSEquipmentProfile::TapSchedule with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | TapChanger | 1..1 | [TapChanger](#UML1998) | A TapSchedule is associated with a TapChanger. |
| 0..\* | DayType | 1..1 | [DayType](#UML1868) | inherited from: [SeasonDayTypeSchedule](#UML1877) |
| 0..\* | Season | 1..1 | [Season](#UML2009) | inherited from: [SeasonDayTypeSchedule](#UML1877) |

### EnergySource

Inheritance path = [EnergyConnection](#UML1920) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 77 shows all attributes of EnergySource.

Table 77 – Attributes of LTDSEquipmentProfile::EnergySource

| name | mult | type | description |
| --- | --- | --- | --- |
| nominalVoltage | 0..1 | [Voltage](#UML54) | Phase-to-phase nominal voltage. |
| pMin | 0..1 | [ActivePower](#UML33) | This is the minimum active power that can be produced by the source. Load sign convention is used, i.e. positive sign means flow out from a TopologicalNode (bus) into the conducting equipment. |
| pMax | 0..1 | [ActivePower](#UML33) | This is the maximum active power that can be produced by the source. Load sign convention is used, i.e. positive sign means flow out from a TopologicalNode (bus) into the conducting equipment. |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 78 – Association ends of LTDSEquipmentProfile::EnergySource with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | EnergySchedulingType | 0..1 | [EnergySchedulingType](#UML1863) | Energy Scheduling Type of an Energy Source. |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### (abstract) Equipment

Inheritance path = [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 79 shows all attributes of Equipment.

Table 79 – Attributes of LTDSEquipmentProfile::Equipment

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | The aggregate flag provides an alternative way of representing an aggregated (equivalent) element. It is applicable in cases when the dedicated classes for equivalent equipment do not have all of the attributes necessary to represent the required level of detail. In case the flag is set to “true” the single instance of equipment represents multiple pieces of equipment that have been modelled together as an aggregate equivalent obtained by a network reduction procedure. Examples would be power transformers or synchronous machines operating in parallel modelled as a single aggregate power transformer or aggregate synchronous machine.  The attribute is not used for EquivalentBranch, EquivalentShunt and EquivalentInjection. |
| normallyInService | 0..1 | [Boolean](#UML57) | Specifies the availability of the equipment under normal operating conditions. True means the equipment is available for topology processing, which determines if the equipment is energized or not. False means that the equipment is treated by network applications as if it is not in the model. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 80 – Association ends of LTDSEquipmentProfile::Equipment with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | Container of this equipment. |

### (abstract) EquipmentContainer

Inheritance path = [ConnectivityNodeContainer](#UML1984) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 81 shows all attributes of EquipmentContainer.

Table 81 – Attributes of LTDSEquipmentProfile::EquipmentContainer

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### EquivalentBranch

Inheritance path = [EquivalentEquipment](#UML1936) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

The class represents equivalent branches. In cases where a transformer phase shift is modelled and the EquivalentBranch is spanning the same nodes, the impedance quantities for the EquivalentBranch shall consider the needed phase shift.

Table 82 shows all attributes of EquivalentBranch.

Table 82 – Attributes of LTDSEquipmentProfile::EquivalentBranch

| name | mult | type | description |
| --- | --- | --- | --- |
| r | 1..1 | [Resistance](#UML50) | Positive sequence series resistance of the reduced branch. |
| r21 | 0..1 | [Resistance](#UML50) | 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](#UML47) | Positive sequence series reactance of the reduced branch. |
| x21 | 0..1 | [Reactance](#UML47) | Reactance from terminal sequence 2 to terminal sequence 1. Used for steady state power flow. This attribute is optional and represents an 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. |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 83 shows all association ends of EquivalentBranch with other classes.

Table 83 – Association ends of LTDSEquipmentProfile::EquivalentBranch with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | EquivalentNetwork | 0..1 | [EquivalentNetwork](#UML1993) | inherited from: [EquivalentEquipment](#UML1936) |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### (abstract) EquivalentEquipment

Inheritance path = [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

The class represents equivalent objects that are the result of a network reduction. The class is the base for equivalent objects of different types.

Table 84 shows all attributes of EquivalentEquipment.

Table 84 – Attributes of LTDSEquipmentProfile::EquivalentEquipment

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 85 shows all association ends of EquivalentEquipment with other classes.

Table 85 – Association ends of LTDSEquipmentProfile::EquivalentEquipment with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | EquivalentNetwork | 0..1 | [EquivalentNetwork](#UML1993) | The equivalent where the reduced model belongs. |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### EquivalentInjection

Inheritance path = [EquivalentEquipment](#UML1936) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

This class represents equivalent injections (generation or load). Voltage regulation is allowed only at the point of connection.

Table 86 shows all attributes of EquivalentInjection.

Table 86 – Attributes of LTDSEquipmentProfile::EquivalentInjection

| name | mult | type | description |
| --- | --- | --- | --- |
| maxP | 0..1 | [ActivePower](#UML33) | Maximum active power of the injection. |
| maxQ | 0..1 | [ReactivePower](#UML48) | Maximum reactive power of the injection. Used for modelling of infeed for load flow exchange. Not used for short circuit modelling. If maxQ and minQ are not used ReactiveCapabilityCurve can be used. |
| minP | 0..1 | [ActivePower](#UML33) | Minimum active power of the injection. |
| minQ | 0..1 | [ReactivePower](#UML48) | Minimum reactive power of the injection. Used for modelling of infeed for load flow exchange. Not used for short circuit modelling. If maxQ and minQ are not used ReactiveCapabilityCurve can be used. |
| regulationCapability | 1..1 | [Boolean](#UML57) | Specifies whether or not the EquivalentInjection has the capability to regulate the local voltage. If true the EquivalentInjection can regulate. If false the EquivalentInjection cannot regulate. ReactiveCapabilityCurve can only be associated with EquivalentInjection if the flag is true. |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 87 shows all association ends of EquivalentInjection with other classes.

Table 87 – Association ends of LTDSEquipmentProfile::EquivalentInjection with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | ReactiveCapabilityCurve | 0..1 | [ReactiveCapabilityCurve](#UML1872) | The reactive capability curve used by this equivalent injection. |
| 0..\* | EquivalentNetwork | 0..1 | [EquivalentNetwork](#UML1993) | inherited from: [EquivalentEquipment](#UML1936) |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### EquivalentNetwork

Inheritance path = [ConnectivityNodeContainer](#UML1984) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A class that groups electrical equivalents, including internal nodes, of a network that has been reduced. 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.

Table 88 shows all attributes of EquivalentNetwork.

Table 88 – Attributes of LTDSEquipmentProfile::EquivalentNetwork

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### EquivalentShunt

Inheritance path = [EquivalentEquipment](#UML1936) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

The class represents equivalent shunts.

Table 89 shows all attributes of EquivalentShunt.

Table 89 – Attributes of LTDSEquipmentProfile::EquivalentShunt

| name | mult | type | description |
| --- | --- | --- | --- |
| b | 1..1 | [Susceptance](#UML53) | Positive sequence shunt susceptance. |
| g | 1..1 | [Conductance](#UML41) | Positive sequence shunt conductance. |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 90 – Association ends of LTDSEquipmentProfile::EquivalentShunt with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | EquivalentNetwork | 0..1 | [EquivalentNetwork](#UML1993) | inherited from: [EquivalentEquipment](#UML1936) |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### ExternalNetworkInjection

Inheritance path = [RegulatingCondEq](#UML1926) : [EnergyConnection](#UML1920) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 91 shows all attributes of ExternalNetworkInjection.

Table 91 – Attributes of LTDSEquipmentProfile::ExternalNetworkInjection

| name | mult | type | description |
| --- | --- | --- | --- |
| governorSCD | 1..1 | [ActivePowerPerFrequency](#UML34) | 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. |
| maxP | 1..1 | [ActivePower](#UML33) | Maximum active power of the injection. |
| maxQ | 1..1 | [ReactivePower](#UML48) | Maximum reactive power limit. It is used for modelling of infeed for load flow exchange and not for short circuit modelling. |
| minP | 1..1 | [ActivePower](#UML33) | Minimum active power of the injection. |
| minQ | 1..1 | [ReactivePower](#UML48) | Minimum reactive power limit. It is used for modelling of infeed for load flow exchange and not for short circuit modelling. |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 92 – Association ends of LTDSEquipmentProfile::ExternalNetworkInjection with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | RegulatingControl | 0..1 | [RegulatingControl](#UML1995) | inherited from: [RegulatingCondEq](#UML1926) |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### FaultIndicator

Inheritance path = [AuxiliaryEquipment](#UML1953) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A FaultIndicator is typically only an indicator (which may or may not be remotely monitored), and not a piece of equipment that actually initiates a protection event. It is used for FLISR (Fault Location, Isolation and Restoration) purposes, assisting with the dispatch of crews to "most likely" part of the network (i.e. assists with determining circuit section where the fault most likely happened).

Table 93 shows all attributes of FaultIndicator.

Table 93 – Attributes of LTDSEquipmentProfile::FaultIndicator

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 94 – Association ends of LTDSEquipmentProfile::FaultIndicator with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Terminal | 1..1 | [Terminal](#UML1888) | inherited from: [AuxiliaryEquipment](#UML1953) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### FossilFuel

Inheritance path = [IdentifiedObject](#UML12)

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

Table 95 shows all attributes of FossilFuel.

Table 95 – Attributes of LTDSEquipmentProfile::FossilFuel

| name | mult | type | description |
| --- | --- | --- | --- |
| fossilFuelType | 1..1 | [FuelType](#UML19) | The type of fossil fuel, such as coal, oil, or gas. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 96 – Association ends of LTDSEquipmentProfile::FossilFuel with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | ThermalGeneratingUnit | 1..1 | [ThermalGeneratingUnit](#UML1975) | A thermal generating unit may have one or more fossil fuels. |

### Fuse

Inheritance path = [Switch](#UML1943) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

An overcurrent protective device with a circuit opening fusible part that is heated and severed by the passage of overcurrent through it. A fuse is considered a switching device because it breaks current.

Table 97 shows all attributes of Fuse.

Table 97 – Attributes of LTDSEquipmentProfile::Fuse

| name | mult | type | description |
| --- | --- | --- | --- |
| normalOpen | 1..1 | [Boolean](#UML57) | inherited from: [Switch](#UML1943) |
| ratedCurrent | 0..1 | [CurrentFlow](#UML42) | inherited from: [Switch](#UML1943) |
| retained | 1..1 | [Boolean](#UML57) | inherited from: [Switch](#UML1943) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 98 – Association ends of LTDSEquipmentProfile::Fuse with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### GeneratingUnit

Inheritance path = [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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.

Table 99 shows all attributes of GeneratingUnit.

Table 99 – Attributes of LTDSEquipmentProfile::GeneratingUnit

| name | mult | type | description |
| --- | --- | --- | --- |
| genControlSource | 0..1 | [GeneratorControlSource](#UML20) | The source of controls for a generating unit. Defines the control status of the generating unit. |
| governorSCD | 0..1 | [PerCent](#UML46) | 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. |
| longPF | 0..1 | [Float](#UML64) | Generating unit long term economic participation factor. |
| maximumAllowableSpinningReserve | 0..1 | [ActivePower](#UML33) | Maximum allowable spinning reserve. Spinning reserve will never be considered greater than this value regardless of the current operating point. |
| maxOperatingP | 1..1 | [ActivePower](#UML33) | This is the maximum operating active power limit the dispatcher can enter for this unit. |
| minOperatingP | 1..1 | [ActivePower](#UML33) | This is the minimum operating active power limit the dispatcher can enter for this unit. |
| nominalP | 0..1 | [ActivePower](#UML33) | 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 to or less than RotatingMachine.ratedS. |
| ratedGrossMaxP | 0..1 | [ActivePower](#UML33) | The unit's gross rated maximum capacity (book value).  The attribute shall be a positive value. |
| ratedGrossMinP | 0..1 | [ActivePower](#UML33) | The gross rated minimum generation level which the unit can safely operate at while delivering power to the transmission grid.  The attribute shall be a positive value. |
| ratedNetMaxP | 0..1 | [ActivePower](#UML33) | The net rated maximum capacity determined by subtracting the auxiliary power used to operate the internal plant machinery from the rated gross maximum capacity.  The attribute shall be a positive value. |
| shortPF | 0..1 | [Float](#UML64) | Generating unit short term economic participation factor. |
| startupCost | 0..1 | [Money](#UML45) | The initial startup cost incurred for each start of the GeneratingUnit. |
| variableCost | 0..1 | [Money](#UML45) | The variable cost component of production per unit of ActivePower. |
| startupTime | 0..1 | [Seconds](#UML52) | Time it takes to get the unit on-line, from the time that the prime mover mechanical power is applied. |
| totalEfficiency | 0..1 | [PerCent](#UML46) | The efficiency of the unit in converting the fuel into electrical energy. |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 100 – Association ends of LTDSEquipmentProfile::GeneratingUnit with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### GeographicalRegion

Inheritance path = [IdentifiedObject](#UML12)

A geographical region of a power system network model.

Table 101 shows all attributes of GeographicalRegion.

Table 101 – Attributes of LTDSEquipmentProfile::GeographicalRegion

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### GrossToNetActivePowerCurve

Inheritance path = [Curve](#UML1869) : [IdentifiedObject](#UML12)

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 modelled, should be treated as non-conforming bus loads. There may be more than one curve, depending on the auxiliary equipment that is in service.

Table 102 shows all attributes of GrossToNetActivePowerCurve.

Table 102 – Attributes of LTDSEquipmentProfile::GrossToNetActivePowerCurve

| name | mult | type | description |
| --- | --- | --- | --- |
| curveStyle | 1..1 | [CurveStyle](#UML18) | inherited from: [Curve](#UML1869) |
| xUnit | 1..1 | [UnitSymbol](#UML30) | inherited from: [Curve](#UML1869) |
| y1Unit | 1..1 | [UnitSymbol](#UML30) | inherited from: [Curve](#UML1869) |
| y2Unit | 0..1 | [UnitSymbol](#UML30) | inherited from: [Curve](#UML1869) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 103 shows all association ends of GrossToNetActivePowerCurve with other classes.

Table 103 – Association ends of LTDSEquipmentProfile::GrossToNetActivePowerCurve with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | GeneratingUnit | 1..1 | [GeneratingUnit](#UML1971) | A generating unit may have a gross active power to net active power curve, describing the losses and auxiliary power requirements of the unit. |

### Ground

Inheritance path = [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 104 shows all attributes of Ground.

Table 104 – Attributes of LTDSEquipmentProfile::Ground

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 105 shows all association ends of Ground with other classes.

Table 105 – Association ends of LTDSEquipmentProfile::Ground with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### GroundDisconnector

Inheritance path = [Switch](#UML1943) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 106 shows all attributes of GroundDisconnector.

Table 106 – Attributes of LTDSEquipmentProfile::GroundDisconnector

| name | mult | type | description |
| --- | --- | --- | --- |
| normalOpen | 1..1 | [Boolean](#UML57) | inherited from: [Switch](#UML1943) |
| ratedCurrent | 0..1 | [CurrentFlow](#UML42) | inherited from: [Switch](#UML1943) |
| retained | 1..1 | [Boolean](#UML57) | inherited from: [Switch](#UML1943) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 107 shows all association ends of GroundDisconnector with other classes.

Table 107 – Association ends of LTDSEquipmentProfile::GroundDisconnector with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### GroundingImpedance

Inheritance path = [EarthFaultCompensator](#UML1917) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A fixed impedance device used for grounding.

Table 108 shows all attributes of GroundingImpedance.

Table 108 – Attributes of LTDSEquipmentProfile::GroundingImpedance

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 109 shows all association ends of GroundingImpedance with other classes.

Table 109 – Association ends of LTDSEquipmentProfile::GroundingImpedance with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### HydroGeneratingUnit

Inheritance path = [GeneratingUnit](#UML1971) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 110 shows all attributes of HydroGeneratingUnit.

Table 110 – Attributes of LTDSEquipmentProfile::HydroGeneratingUnit

| name | mult | type | description |
| --- | --- | --- | --- |
| energyConversionCapability | 0..1 | [HydroEnergyConversionKind](#UML21) | Energy conversion capability for generating. |
| dropHeight | 0..1 | [Length](#UML44) | The height water drops from the reservoir mid-point to the turbine. |
| turbineType | 0..1 | [HydroTurbineKind](#UML23) | Type of turbine. |
| genControlSource | 0..1 | [GeneratorControlSource](#UML20) | inherited from: [GeneratingUnit](#UML1971) |
| governorSCD | 0..1 | [PerCent](#UML46) | inherited from: [GeneratingUnit](#UML1971) |
| longPF | 0..1 | [Float](#UML64) | inherited from: [GeneratingUnit](#UML1971) |
| maximumAllowableSpinningReserve | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| maxOperatingP | 1..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| minOperatingP | 1..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| nominalP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| ratedGrossMaxP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| ratedGrossMinP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| ratedNetMaxP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| shortPF | 0..1 | [Float](#UML64) | inherited from: [GeneratingUnit](#UML1971) |
| startupCost | 0..1 | [Money](#UML45) | inherited from: [GeneratingUnit](#UML1971) |
| variableCost | 0..1 | [Money](#UML45) | inherited from: [GeneratingUnit](#UML1971) |
| startupTime | 0..1 | [Seconds](#UML52) | inherited from: [GeneratingUnit](#UML1971) |
| totalEfficiency | 0..1 | [PerCent](#UML46) | inherited from: [GeneratingUnit](#UML1971) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 111 shows all association ends of HydroGeneratingUnit with other classes.

Table 111 – Association ends of LTDSEquipmentProfile::HydroGeneratingUnit with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | HydroPowerPlant | 0..1 | [HydroPowerPlant](#UML1903) | The hydro generating unit belongs to a hydro power plant. |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### HydroPowerPlant

Inheritance path = [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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.

Table 112 shows all attributes of HydroPowerPlant.

Table 112 – Attributes of LTDSEquipmentProfile::HydroPowerPlant

| name | mult | type | description |
| --- | --- | --- | --- |
| hydroPlantStorageType | 1..1 | [HydroPlantStorageKind](#UML22) | The type of hydro power plant water storage. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### HydroPump

Inheritance path = [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 113 shows all attributes of HydroPump.

Table 113 – Attributes of LTDSEquipmentProfile::HydroPump

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 114 – Association ends of LTDSEquipmentProfile::HydroPump with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..1 | RotatingMachine | 1..1 | [RotatingMachine](#UML1929) | 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. |
| 0..\* | HydroPowerPlant | 0..1 | [HydroPowerPlant](#UML1903) | The hydro pump may be a member of a pumped storage plant or a pump for distributing water. |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### (abstract) IdentifiedObject root class

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

Table 115 shows all attributes of IdentifiedObject.

Table 115 – Attributes of LTDSEquipmentProfile::IdentifiedObject

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | 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. |
| mRID | 1..1 | [String](#UML67) | Master resource identifier issued by a model authority. The mRID is unique within an exchange context. Global uniqueness is easily achieved by using a UUID, as specified in RFC 4122, for the mRID. The use of UUID is strongly recommended.  For CIMXML data files in RDF syntax conforming to IEC 61970-552, the mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements. |
| name | 1..1 | [String](#UML67) | The name is any free human readable and possibly non unique text naming the object. |

### Jumper

Inheritance path = [Switch](#UML1943) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A short section of conductor with negligible impedance which can be manually removed and replaced if the circuit is de-energized. Note that zero-impedance branches can potentially be modelled by other equipment types.

Table 116 shows all attributes of Jumper.

Table 116 – Attributes of LTDSEquipmentProfile::Jumper

| name | mult | type | description |
| --- | --- | --- | --- |
| normalOpen | 1..1 | [Boolean](#UML57) | inherited from: [Switch](#UML1943) |
| ratedCurrent | 0..1 | [CurrentFlow](#UML42) | inherited from: [Switch](#UML1943) |
| retained | 1..1 | [Boolean](#UML57) | inherited from: [Switch](#UML1943) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 117 shows all association ends of Jumper with other classes.

Table 117 – Association ends of LTDSEquipmentProfile::Jumper with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### Junction

Inheritance path = [Connector](#UML1914) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 118 shows all attributes of Junction.

Table 118 – Attributes of LTDSEquipmentProfile::Junction

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 119 shows all association ends of Junction with other classes.

Table 119 – Association ends of LTDSEquipmentProfile::Junction with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### Line

Inheritance path = [EquipmentContainer](#UML1985) : [ConnectivityNodeContainer](#UML1984) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 120 shows all attributes of Line.

Table 120 – Attributes of LTDSEquipmentProfile::Line

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 121 shows all association ends of Line with other classes.

Table 121 – Association ends of LTDSEquipmentProfile::Line with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Region | 0..1 | [SubGeographicalRegion](#UML2010) | The sub-geographical region of the line. |

### LinearShuntCompensator

Inheritance path = [ShuntCompensator](#UML1932) : [RegulatingCondEq](#UML1926) : [EnergyConnection](#UML1920) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 122 shows all attributes of LinearShuntCompensator.

Table 122 – Attributes of LTDSEquipmentProfile::LinearShuntCompensator

| name | mult | type | description |
| --- | --- | --- | --- |
| bPerSection | 1..1 | [Susceptance](#UML53) | Positive sequence shunt (charging) susceptance per section. |
| gPerSection | 1..1 | [Conductance](#UML41) | Positive sequence shunt (charging) conductance per section. |
| aVRDelay | 0..1 | [Seconds](#UML52) | inherited from: [ShuntCompensator](#UML1932) |
| grounded | 0..1 | [Boolean](#UML57) | inherited from: [ShuntCompensator](#UML1932) |
| maximumSections | 1..1 | [Integer](#UML65) | inherited from: [ShuntCompensator](#UML1932) |
| nomU | 1..1 | [Voltage](#UML54) | inherited from: [ShuntCompensator](#UML1932) |
| normalSections | 1..1 | [Integer](#UML65) | inherited from: [ShuntCompensator](#UML1932) |
| voltageSensitivity | 0..1 | [VoltagePerReactivePower](#UML55) | inherited from: [ShuntCompensator](#UML1932) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 123 – Association ends of LTDSEquipmentProfile::LinearShuntCompensator with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | RegulatingControl | 0..1 | [RegulatingControl](#UML1995) | inherited from: [RegulatingCondEq](#UML1926) |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### LoadArea

Inheritance path = [EnergyArea](#UML1865) : [IdentifiedObject](#UML12)

The class is the root or first level in a hierarchical structure for grouping of loads for the purpose of load flow load scaling.

Table 124 shows all attributes of LoadArea.

Table 124 – Attributes of LTDSEquipmentProfile::LoadArea

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### LoadBreakSwitch

Inheritance path = [ProtectedSwitch](#UML1944) : [Switch](#UML1943) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 125 shows all attributes of LoadBreakSwitch.

Table 125 – Attributes of LTDSEquipmentProfile::LoadBreakSwitch

| name | mult | type | description |
| --- | --- | --- | --- |
| normalOpen | 1..1 | [Boolean](#UML57) | inherited from: [Switch](#UML1943) |
| ratedCurrent | 0..1 | [CurrentFlow](#UML42) | inherited from: [Switch](#UML1943) |
| retained | 1..1 | [Boolean](#UML57) | inherited from: [Switch](#UML1943) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 126 – Association ends of LTDSEquipmentProfile::LoadBreakSwitch with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### (abstract) LoadGroup

Inheritance path = [IdentifiedObject](#UML12)

The class is the third level in a hierarchical structure for grouping of loads for the purpose of load flow load scaling.

Table 127 shows all attributes of LoadGroup.

Table 127 – Attributes of LTDSEquipmentProfile::LoadGroup

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 128 – Association ends of LTDSEquipmentProfile::LoadGroup with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 1..\* | SubLoadArea | 1..1 | [SubLoadArea](#UML1867) | The SubLoadArea where the Loadgroup belongs. |

### LoadResponseCharacteristic

Inheritance path = [IdentifiedObject](#UML12)

Models the characteristic response of the load demand due to changes in system conditions such as voltage and frequency. It is not related to demand response.

If LoadResponseCharacteristic.exponentModel is True, the exponential voltage or frequency dependent models are specified and used as to calculate active and reactive power components of the load model.

The equations to calculate active and reactive power components of the load model are internal to the power flow calculation, hence they use different quantities depending on the use case of the data exchange.

The equations for exponential voltage dependent load model injected power are:

pInjection= Pnominal\* (Voltage/cim:BaseVoltage.nominalVoltage) \*\* cim:LoadResponseCharacteristic.pVoltageExponent

qInjection= Qnominal\* (Voltage/cim:BaseVoltage.nominalVoltage) \*\* cim:LoadResponseCharacteristic.qVoltageExponent

Where:

1) \* means "multiply" and \*\* is "raised to power of";

2) Pnominal and Qnominal represent the active power and reactive power at nominal voltage as any load described by the voltage exponential model shall be given at nominal voltage. This means that EnergyConsumer.p and EnergyConsumer.q are at nominal voltage.

3) After power flow is solved:

-pInjection and qInjection correspond to SvPowerflow.p and SvPowerflow.q respectively.

- Voltage corresponds to SvVoltage.v at the TopologicalNode where the load is connected.

Table 129 shows all attributes of LoadResponseCharacteristic.

Table 129 – Attributes of LTDSEquipmentProfile::LoadResponseCharacteristic

| name | mult | type | description |
| --- | --- | --- | --- |
| exponentModel | 1..1 | [Boolean](#UML57) | 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  - pFrequencyExponent  - qFrequencyExponent.  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 | [Float](#UML64) | Portion of active power load modelled as constant current. |
| pConstantImpedance | 0..1 | [Float](#UML64) | Portion of active power load modelled as constant impedance. |
| pConstantPower | 0..1 | [Float](#UML64) | Portion of active power load modelled as constant power. |
| pFrequencyExponent | 0..1 | [Float](#UML64) | Exponent of per unit frequency effecting active power. |
| pVoltageExponent | 0..1 | [Float](#UML64) | Exponent of per unit voltage effecting real power. |
| qConstantCurrent | 0..1 | [Float](#UML64) | Portion of reactive power load modelled as constant current. |
| qConstantImpedance | 0..1 | [Float](#UML64) | Portion of reactive power load modelled as constant impedance. |
| qConstantPower | 0..1 | [Float](#UML64) | Portion of reactive power load modelled as constant power. |
| qFrequencyExponent | 0..1 | [Float](#UML64) | Exponent of per unit frequency effecting reactive power. |
| qVoltageExponent | 0..1 | [Float](#UML64) | Exponent of per unit voltage effecting reactive power. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### NonConformLoad

Inheritance path = [EnergyConsumer](#UML1921) : [EnergyConnection](#UML1920) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

NonConformLoad represents loads that do not follow a daily load change pattern and whose changes are not correlated with the daily load change pattern.

Table 130 shows all attributes of NonConformLoad.

Table 130 – Attributes of LTDSEquipmentProfile::NonConformLoad

| name | mult | type | description |
| --- | --- | --- | --- |
| pfixed | 0..1 | [ActivePower](#UML33) | inherited from: [EnergyConsumer](#UML1921) |
| pfixedPct | 0..1 | [PerCent](#UML46) | inherited from: [EnergyConsumer](#UML1921) |
| qfixed | 0..1 | [ReactivePower](#UML48) | inherited from: [EnergyConsumer](#UML1921) |
| qfixedPct | 0..1 | [PerCent](#UML46) | inherited from: [EnergyConsumer](#UML1921) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 131 shows all association ends of NonConformLoad with other classes.

Table 131 – Association ends of LTDSEquipmentProfile::NonConformLoad with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 1..\* | LoadGroup | 1..1 | [NonConformLoadGroup](#UML1892) | Group of this ConformLoad. |
| 0..\* | LoadResponse | 0..1 | [LoadResponseCharacteristic](#UML1893) | inherited from: [EnergyConsumer](#UML1921) |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### NonConformLoadGroup

Inheritance path = [LoadGroup](#UML1890) : [IdentifiedObject](#UML12)

Loads that do not follow a daily and seasonal load variation pattern.

Table 132 shows all attributes of NonConformLoadGroup.

Table 132 – Attributes of LTDSEquipmentProfile::NonConformLoadGroup

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 133 shows all association ends of NonConformLoadGroup with other classes.

Table 133 – Association ends of LTDSEquipmentProfile::NonConformLoadGroup with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 1..\* | SubLoadArea | 1..1 | [SubLoadArea](#UML1867) | inherited from: [LoadGroup](#UML1890) |

### NonConformLoadSchedule

Inheritance path = [SeasonDayTypeSchedule](#UML1877) : [RegularIntervalSchedule](#UML1876) : [BasicIntervalSchedule](#UML1875) : [IdentifiedObject](#UML12)

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 modelled).

Table 134 shows all attributes of NonConformLoadSchedule.

Table 134 – Attributes of LTDSEquipmentProfile::NonConformLoadSchedule

| name | mult | type | description |
| --- | --- | --- | --- |
| timeStep | 1..1 | [Seconds](#UML52) | inherited from: [RegularIntervalSchedule](#UML1876) |
| endTime | 1..1 | [DateTime](#UML63) | inherited from: [RegularIntervalSchedule](#UML1876) |
| startTime | 1..1 | [DateTime](#UML63) | inherited from: [BasicIntervalSchedule](#UML1875) |
| value1Unit | 1..1 | [UnitSymbol](#UML30) | inherited from: [BasicIntervalSchedule](#UML1875) |
| value2Unit | 0..1 | [UnitSymbol](#UML30) | inherited from: [BasicIntervalSchedule](#UML1875) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 135 shows all association ends of NonConformLoadSchedule with other classes.

Table 135 – Association ends of LTDSEquipmentProfile::NonConformLoadSchedule with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | NonConformLoadGroup | 1..1 | [NonConformLoadGroup](#UML1892) | The NonConformLoadGroup where the NonConformLoadSchedule belongs. |
| 0..\* | DayType | 1..1 | [DayType](#UML1868) | inherited from: [SeasonDayTypeSchedule](#UML1877) |
| 0..\* | Season | 1..1 | [Season](#UML2009) | inherited from: [SeasonDayTypeSchedule](#UML1877) |

### NonlinearShuntCompensator

Inheritance path = [ShuntCompensator](#UML1932) : [RegulatingCondEq](#UML1926) : [EnergyConnection](#UML1920) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A non linear shunt compensator has bank or section admittance values that differ. The attributes g, b, g0 and b0 of the associated NonlinearShuntCompensatorPoint describe the total conductance and admittance of a NonlinearShuntCompensatorPoint at a section number specified by NonlinearShuntCompensatorPoint.sectionNumber.

Table 136 shows all attributes of NonlinearShuntCompensator.

Table 136 – Attributes of LTDSEquipmentProfile::NonlinearShuntCompensator

| name | mult | type | description |
| --- | --- | --- | --- |
| aVRDelay | 0..1 | [Seconds](#UML52) | inherited from: [ShuntCompensator](#UML1932) |
| grounded | 0..1 | [Boolean](#UML57) | inherited from: [ShuntCompensator](#UML1932) |
| maximumSections | 1..1 | [Integer](#UML65) | inherited from: [ShuntCompensator](#UML1932) |
| nomU | 1..1 | [Voltage](#UML54) | inherited from: [ShuntCompensator](#UML1932) |
| normalSections | 1..1 | [Integer](#UML65) | inherited from: [ShuntCompensator](#UML1932) |
| voltageSensitivity | 0..1 | [VoltagePerReactivePower](#UML55) | inherited from: [ShuntCompensator](#UML1932) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 137 – Association ends of LTDSEquipmentProfile::NonlinearShuntCompensator with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | RegulatingControl | 0..1 | [RegulatingControl](#UML1995) | inherited from: [RegulatingCondEq](#UML1926) |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### NonlinearShuntCompensatorPoint root class

A non linear shunt compensator bank or section admittance value. The number of NonlinearShuntCompenstorPoint instances associated with a NonlinearShuntCompensator shall be equal to ShuntCompensator.maximumSections. ShuntCompensator.sections shall only be set to one of the NonlinearShuntCompenstorPoint.sectionNumber. There is no interpolation between NonlinearShuntCompenstorPoint-s.

Table 138 shows all attributes of NonlinearShuntCompensatorPoint.

Table 138 – Attributes of LTDSEquipmentProfile::NonlinearShuntCompensatorPoint

| name | mult | type | description |
| --- | --- | --- | --- |
| b | 1..1 | [Susceptance](#UML53) | Positive sequence shunt (charging) susceptance per section. |
| g | 1..1 | [Conductance](#UML41) | Positive sequence shunt (charging) conductance per section. |
| sectionNumber | 1..1 | [Integer](#UML65) | The number of the section. |

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

Table 139 – Association ends of LTDSEquipmentProfile::NonlinearShuntCompensatorPoint with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 1..\* | NonlinearShuntCompensator | 1..1 | [NonlinearShuntCompensator](#UML1933) | Non-linear shunt compensator owning this point. |

### NuclearGeneratingUnit

Inheritance path = [GeneratingUnit](#UML1971) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A nuclear generating unit.

Table 140 shows all attributes of NuclearGeneratingUnit.

Table 140 – Attributes of LTDSEquipmentProfile::NuclearGeneratingUnit

| name | mult | type | description |
| --- | --- | --- | --- |
| genControlSource | 0..1 | [GeneratorControlSource](#UML20) | inherited from: [GeneratingUnit](#UML1971) |
| governorSCD | 0..1 | [PerCent](#UML46) | inherited from: [GeneratingUnit](#UML1971) |
| longPF | 0..1 | [Float](#UML64) | inherited from: [GeneratingUnit](#UML1971) |
| maximumAllowableSpinningReserve | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| maxOperatingP | 1..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| minOperatingP | 1..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| nominalP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| ratedGrossMaxP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| ratedGrossMinP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| ratedNetMaxP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| shortPF | 0..1 | [Float](#UML64) | inherited from: [GeneratingUnit](#UML1971) |
| startupCost | 0..1 | [Money](#UML45) | inherited from: [GeneratingUnit](#UML1971) |
| variableCost | 0..1 | [Money](#UML45) | inherited from: [GeneratingUnit](#UML1971) |
| startupTime | 0..1 | [Seconds](#UML52) | inherited from: [GeneratingUnit](#UML1971) |
| totalEfficiency | 0..1 | [PerCent](#UML46) | inherited from: [GeneratingUnit](#UML1971) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 141 – Association ends of LTDSEquipmentProfile::NuclearGeneratingUnit with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### (abstract) OperationalLimit

Inheritance path = [IdentifiedObject](#UML12)

A value and normal value associated with a specific kind of limit.

The sub class value and normalValue attributes vary inversely to the associated OperationalLimitType.acceptableDuration (acceptableDuration for short).

If a particular piece of equipment has multiple operational limits of the same kind (apparent power, current, etc.), the limit with the greatest acceptableDuration shall have the smallest limit value and the limit with the smallest acceptableDuration shall have the largest limit value. Note: A large current can only be allowed to flow through a piece of equipment for a short duration without causing damage, but a lesser current can be allowed to flow for a longer duration.

Table 142 shows all attributes of OperationalLimit.

Table 142 – Attributes of LTDSEquipmentProfile::OperationalLimit

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 143 – Association ends of LTDSEquipmentProfile::OperationalLimit with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | OperationalLimitType | 1..1 | [OperationalLimitType](#UML1900) | The limit type associated with this limit. |
| 0..\* | OperationalLimitSet | 1..1 | [OperationalLimitSet](#UML1899) | The limit set to which the limit values belong. |

### OperationalLimitSet

Inheritance path = [IdentifiedObject](#UML12)

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.

Table 144 shows all attributes of OperationalLimitSet.

Table 144 – Attributes of LTDSEquipmentProfile::OperationalLimitSet

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 145 – Association ends of LTDSEquipmentProfile::OperationalLimitSet with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Terminal | 1..1 | [ACDCTerminal](#UML1884) | The terminal where the operational limit set apply. |
| 0..\* | Equipment | 0..1 | [Equipment](#UML1906) | The equipment to which the limit set applies. |

### OperationalLimitType

Inheritance path = [IdentifiedObject](#UML12)

The operational meaning of a category of limits.

Table 146 shows all attributes of OperationalLimitType.

Table 146 – Attributes of LTDSEquipmentProfile::OperationalLimitType

| name | mult | type | description |
| --- | --- | --- | --- |
| acceptableDuration | 0..1 | [Seconds](#UML52) | The nominal acceptable duration of the limit. Limits are commonly expressed in terms of the 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. The attribute has meaning only if the flag isInfiniteDuration is set to false, hence it shall not be exchanged when isInfiniteDuration is set to true. |
| direction | 1..1 | [OperationalLimitDirectionKind](#UML25) | The direction of the limit. |
| isInfiniteDuration | 1..1 | [Boolean](#UML57) | Defines if the operational limit type has infinite duration. If true, the limit has infinite duration. If false, the limit has definite duration which is defined by the attribute acceptableDuration. |
| kind | 1..1 | [LimitKind](#UML24) | (European) Types of limits defined in the ENTSO-E Operational Handbook Policy 3. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### PetersenCoil

Inheritance path = [EarthFaultCompensator](#UML1917) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 147 shows all attributes of PetersenCoil.

Table 147 – Attributes of LTDSEquipmentProfile::PetersenCoil

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 148 shows all association ends of PetersenCoil with other classes.

Table 148 – Association ends of LTDSEquipmentProfile::PetersenCoil with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### (abstract) PhaseTapChanger

Inheritance path = [TapChanger](#UML1998) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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.

Table 149 shows all attributes of PhaseTapChanger.

Table 149 – Attributes of LTDSEquipmentProfile::PhaseTapChanger

| name | mult | type | description |
| --- | --- | --- | --- |
| highStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| lowStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| ltcFlag | 1..1 | [Boolean](#UML57) | inherited from: [TapChanger](#UML1998) |
| neutralStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| neutralU | 1..1 | [Voltage](#UML54) | inherited from: [TapChanger](#UML1998) |
| normalStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 150 shows all association ends of PhaseTapChanger with other classes.

Table 150 – Association ends of LTDSEquipmentProfile::PhaseTapChanger with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..1 | TransformerEnd | 1..1 | [TransformerEnd](#UML2012) | Transformer end to which this phase tap changer belongs. |
| 1..\* | TapChangerControl | 0..1 | [TapChangerControl](#UML1996) | inherited from: [TapChanger](#UML1998) |

### PhaseTapChangerAsymmetrical

Inheritance path = [PhaseTapChangerNonLinear](#UML2002) : [PhaseTapChanger](#UML2000) : [TapChanger](#UML1998) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

Describes the tap model for an asymmetrical phase shifting transformer in which the difference voltage vector adds to the in-phase winding. The out-of-phase winding is the transformer end where the tap changer is located. The angle between the in-phase and out-of-phase windings is named the winding connection angle. The phase shift depends on both the difference voltage magnitude and the winding connection angle.

Table 151 shows all attributes of PhaseTapChangerAsymmetrical.

Table 151 – Attributes of LTDSEquipmentProfile::PhaseTapChangerAsymmetrical

| name | mult | type | description |
| --- | --- | --- | --- |
| windingConnectionAngle | 1..1 | [AngleDegrees](#UML35) | 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 symmetrical transformer. The attribute can only be multiples of 30 degrees. The allowed range is -150 degrees to 150 degrees excluding 0. |
| voltageStepIncrement | 1..1 | [PerCent](#UML46) | inherited from: [PhaseTapChangerNonLinear](#UML2002) |
| xMax | 1..1 | [Reactance](#UML47) | inherited from: [PhaseTapChangerNonLinear](#UML2002) |
| xMin | 1..1 | [Reactance](#UML47) | (deprecated) inherited from: [PhaseTapChangerNonLinear](#UML2002) |
| highStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| lowStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| ltcFlag | 1..1 | [Boolean](#UML57) | inherited from: [TapChanger](#UML1998) |
| neutralStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| neutralU | 1..1 | [Voltage](#UML54) | inherited from: [TapChanger](#UML1998) |
| normalStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 152 shows all association ends of PhaseTapChangerAsymmetrical with other classes.

Table 152 – Association ends of LTDSEquipmentProfile::PhaseTapChangerAsymmetrical with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..1 | TransformerEnd | 1..1 | [TransformerEnd](#UML2012) | inherited from: [PhaseTapChanger](#UML2000) |
| 1..\* | TapChangerControl | 0..1 | [TapChangerControl](#UML1996) | inherited from: [TapChanger](#UML1998) |

### PhaseTapChangerLinear

Inheritance path = [PhaseTapChanger](#UML2000) : [TapChanger](#UML1998) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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 stepPhaseShiftIncrement times the tap position.

The voltage magnitude of both sides is the same.

Table 153 shows all attributes of PhaseTapChangerLinear.

Table 153 – Attributes of LTDSEquipmentProfile::PhaseTapChangerLinear

| name | mult | type | description |
| --- | --- | --- | --- |
| stepPhaseShiftIncrement | 1..1 | [AngleDegrees](#UML35) | Phase shift per step position. A positive value indicates a positive angle variation from the Terminal at the PowerTransformerEnd, where the TapChanger is located, into the 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](#UML47) | The reactance depends on the tap position according to a "u" shaped curve. The maximum reactance (xMax) appears at the low and high tap positions. Depending on the “u” curve the attribute can be either higher or lower than PowerTransformerEnd.x. |
| xMin | 1..1 | [Reactance](#UML47) | (deprecated) The reactance depends on the tap position according to a "u" shaped curve. The minimum reactance (xMin) appears at the mid tap position. PowerTransformerEnd.x shall be consistent with PhaseTapChangerLinear.xMin and PhaseTapChangerNonLinear.xMin. In case of inconsistency, PowerTransformerEnd.x shall be used. |
| highStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| lowStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| ltcFlag | 1..1 | [Boolean](#UML57) | inherited from: [TapChanger](#UML1998) |
| neutralStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| neutralU | 1..1 | [Voltage](#UML54) | inherited from: [TapChanger](#UML1998) |
| normalStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 154 shows all association ends of PhaseTapChangerLinear with other classes.

Table 154 – Association ends of LTDSEquipmentProfile::PhaseTapChangerLinear with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..1 | TransformerEnd | 1..1 | [TransformerEnd](#UML2012) | inherited from: [PhaseTapChanger](#UML2000) |
| 1..\* | TapChangerControl | 0..1 | [TapChangerControl](#UML1996) | inherited from: [TapChanger](#UML1998) |

### (abstract) PhaseTapChangerNonLinear

Inheritance path = [PhaseTapChanger](#UML2000) : [TapChanger](#UML1998) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

The non-linear phase tap changer describes the non-linear behaviour 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.

Table 155 shows all attributes of PhaseTapChangerNonLinear.

Table 155 – Attributes of LTDSEquipmentProfile::PhaseTapChangerNonLinear

| name | mult | type | description |
| --- | --- | --- | --- |
| voltageStepIncrement | 1..1 | [PerCent](#UML46) | The voltage step increment on the out of phase winding (the PowerTransformerEnd where the TapChanger is located) specified in percent of rated voltage of the PowerTransformerEnd. A positive value means a positive voltage variation from the Terminal at the PowerTransformerEnd, where the TapChanger is located, into the transformer.  When the increment is negative, the voltage decreases when the tap step increases. |
| xMax | 1..1 | [Reactance](#UML47) | The reactance depends on the tap position according to a "u" shaped curve. The maximum reactance (xMax) appears at the low and high tap positions. Depending on the “u” curve the attribute can be either higher or lower than PowerTransformerEnd.x. |
| xMin | 1..1 | [Reactance](#UML47) | (deprecated) The reactance depend on the tap position according to a "u" shaped curve. The minimum reactance (xMin) appear at the mid tap position. PowerTransformerEnd.x shall be consistent with PhaseTapChangerLinear.xMin and PhaseTapChangerNonLinear.xMin. In case of inconsistency, PowerTransformerEnd.x shall be used. |
| highStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| lowStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| ltcFlag | 1..1 | [Boolean](#UML57) | inherited from: [TapChanger](#UML1998) |
| neutralStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| neutralU | 1..1 | [Voltage](#UML54) | inherited from: [TapChanger](#UML1998) |
| normalStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 156 shows all association ends of PhaseTapChangerNonLinear with other classes.

Table 156 – Association ends of LTDSEquipmentProfile::PhaseTapChangerNonLinear with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..1 | TransformerEnd | 1..1 | [TransformerEnd](#UML2012) | inherited from: [PhaseTapChanger](#UML2000) |
| 1..\* | TapChangerControl | 0..1 | [TapChangerControl](#UML1996) | inherited from: [TapChanger](#UML1998) |

### PhaseTapChangerSymmetrical

Inheritance path = [PhaseTapChangerNonLinear](#UML2002) : [PhaseTapChanger](#UML2000) : [TapChanger](#UML1998) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

Describes a symmetrical phase shifting transformer tap model in which the voltage magnitude of both sides is the same. 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.

Table 157 shows all attributes of PhaseTapChangerSymmetrical.

Table 157 – Attributes of LTDSEquipmentProfile::PhaseTapChangerSymmetrical

| name | mult | type | description |
| --- | --- | --- | --- |
| voltageStepIncrement | 1..1 | [PerCent](#UML46) | inherited from: [PhaseTapChangerNonLinear](#UML2002) |
| xMax | 1..1 | [Reactance](#UML47) | inherited from: [PhaseTapChangerNonLinear](#UML2002) |
| xMin | 1..1 | [Reactance](#UML47) | (deprecated) inherited from: [PhaseTapChangerNonLinear](#UML2002) |
| highStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| lowStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| ltcFlag | 1..1 | [Boolean](#UML57) | inherited from: [TapChanger](#UML1998) |
| neutralStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| neutralU | 1..1 | [Voltage](#UML54) | inherited from: [TapChanger](#UML1998) |
| normalStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 158 shows all association ends of PhaseTapChangerSymmetrical with other classes.

Table 158 – Association ends of LTDSEquipmentProfile::PhaseTapChangerSymmetrical with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..1 | TransformerEnd | 1..1 | [TransformerEnd](#UML2012) | inherited from: [PhaseTapChanger](#UML2000) |
| 1..\* | TapChangerControl | 0..1 | [TapChangerControl](#UML1996) | inherited from: [TapChanger](#UML1998) |

### PhaseTapChangerTable

Inheritance path = [IdentifiedObject](#UML12)

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

Table 159 shows all attributes of PhaseTapChangerTable.

Table 159 – Attributes of LTDSEquipmentProfile::PhaseTapChangerTable

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### PhaseTapChangerTablePoint

Inheritance path = [TapChangerTablePoint](#UML15)

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

Table 160 shows all attributes of PhaseTapChangerTablePoint.

Table 160 – Attributes of LTDSEquipmentProfile::PhaseTapChangerTablePoint

| name | mult | type | description |
| --- | --- | --- | --- |
| angle | 1..1 | [AngleDegrees](#UML35) | The angle difference in degrees. A positive value indicates a positive angle variation from the Terminal at the PowerTransformerEnd, where the TapChanger is located, into the transformer. |
| b | 0..1 | [PerCent](#UML46) | inherited from: [TapChangerTablePoint](#UML15) |
| g | 0..1 | [PerCent](#UML46) | inherited from: [TapChangerTablePoint](#UML15) |
| r | 0..1 | [PerCent](#UML46) | inherited from: [TapChangerTablePoint](#UML15) |
| ratio | 0..1 | [Float](#UML64) | inherited from: [TapChangerTablePoint](#UML15) |
| step | 1..1 | [Integer](#UML65) | inherited from: [TapChangerTablePoint](#UML15) |
| x | 0..1 | [PerCent](#UML46) | inherited from: [TapChangerTablePoint](#UML15) |

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

Table 161 – Association ends of LTDSEquipmentProfile::PhaseTapChangerTablePoint with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 1..\* | PhaseTapChangerTable | 1..1 | [PhaseTapChangerTable](#UML1901) | The table of this point. |

### PhaseTapChangerTabular

Inheritance path = [PhaseTapChanger](#UML2000) : [TapChanger](#UML1998) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

Describes a tap changer with a table defining the relation between the tap step and the phase angle difference across the transformer.

Table 162 shows all attributes of PhaseTapChangerTabular.

Table 162 – Attributes of LTDSEquipmentProfile::PhaseTapChangerTabular

| name | mult | type | description |
| --- | --- | --- | --- |
| highStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| lowStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| ltcFlag | 1..1 | [Boolean](#UML57) | inherited from: [TapChanger](#UML1998) |
| neutralStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| neutralU | 1..1 | [Voltage](#UML54) | inherited from: [TapChanger](#UML1998) |
| normalStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 163 shows all association ends of PhaseTapChangerTabular with other classes.

Table 163 – Association ends of LTDSEquipmentProfile::PhaseTapChangerTabular with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | PhaseTapChangerTable | 1..1 | [PhaseTapChangerTable](#UML1901) | The phase tap changer table for this phase tap changer. |
| 0..1 | TransformerEnd | 1..1 | [TransformerEnd](#UML2012) | inherited from: [PhaseTapChanger](#UML2000) |
| 1..\* | TapChangerControl | 0..1 | [TapChangerControl](#UML1996) | inherited from: [TapChanger](#UML1998) |

### PhotoVoltaicUnit

Inheritance path = [PowerElectronicsUnit](#UML1978) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A photovoltaic device or an aggregation of such devices.

Table 164 shows all attributes of PhotoVoltaicUnit.

Table 164 – Attributes of LTDSEquipmentProfile::PhotoVoltaicUnit

| name | mult | type | description |
| --- | --- | --- | --- |
| maxP | 0..1 | [ActivePower](#UML33) | inherited from: [PowerElectronicsUnit](#UML1978) |
| minP | 0..1 | [ActivePower](#UML33) | inherited from: [PowerElectronicsUnit](#UML1978) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 165 shows all association ends of PhotoVoltaicUnit with other classes.

Table 165 – Association ends of LTDSEquipmentProfile::PhotoVoltaicUnit with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### PostLineSensor

Inheritance path = [Sensor](#UML1955) : [AuxiliaryEquipment](#UML1953) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A sensor used mainly in overhead distribution networks as the source of both current and voltage measurements.

Table 166 shows all attributes of PostLineSensor.

Table 166 – Attributes of LTDSEquipmentProfile::PostLineSensor

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 167 shows all association ends of PostLineSensor with other classes.

Table 167 – Association ends of LTDSEquipmentProfile::PostLineSensor with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Terminal | 1..1 | [Terminal](#UML1888) | inherited from: [AuxiliaryEquipment](#UML1953) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### PotentialTransformer

Inheritance path = [Sensor](#UML1955) : [AuxiliaryEquipment](#UML1953) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

Instrument transformer (also known as Voltage Transformer) used to measure electrical qualities of the circuit that is being protected and/or monitored. Typically used as voltage transducer for the purpose of metering, protection, or sometimes auxiliary substation supply. A typical secondary voltage rating would be 120V.

Table 168 shows all attributes of PotentialTransformer.

Table 168 – Attributes of LTDSEquipmentProfile::PotentialTransformer

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 169 shows all association ends of PotentialTransformer with other classes.

Table 169 – Association ends of LTDSEquipmentProfile::PotentialTransformer with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Terminal | 1..1 | [Terminal](#UML1888) | inherited from: [AuxiliaryEquipment](#UML1953) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### PowerElectronicsConnection

Inheritance path = [RegulatingCondEq](#UML1926) : [EnergyConnection](#UML1920) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A connection to the AC network for energy production or consumption that uses power electronics rather than rotating machines.

Table 170 shows all attributes of PowerElectronicsConnection.

Table 170 – Attributes of LTDSEquipmentProfile::PowerElectronicsConnection

| name | mult | type | description |
| --- | --- | --- | --- |
| maxQ | 0..1 | [ReactivePower](#UML48) | Maximum reactive power limit. This is the maximum (nameplate) limit for the unit. |
| minQ | 0..1 | [ReactivePower](#UML48) | Minimum reactive power limit for the unit. This is the minimum (nameplate) limit for the unit. |
| ratedS | 0..1 | [ApparentPower](#UML40) | Nameplate apparent power rating for the unit.  The attribute shall have a positive value. |
| ratedU | 0..1 | [Voltage](#UML54) | Rated voltage (nameplate data, Ur in IEC 60909-0). It is primarily used for short circuit data exchange according to IEC 60909.  The attribute shall be a positive value. |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 171 – Association ends of LTDSEquipmentProfile::PowerElectronicsConnection with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 1..1 | PowerElectronicsUnit | 0..1 | [PowerElectronicsUnit](#UML1978) | An AC network connection may have several power electronics units connecting through it. |
| 0..\* | RegulatingControl | 0..1 | [RegulatingControl](#UML1995) | inherited from: [RegulatingCondEq](#UML1926) |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### (abstract) PowerElectronicsUnit

Inheritance path = [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A generating unit or battery or aggregation that connects to the AC network using power electronics rather than rotating machines.

Table 172 shows all attributes of PowerElectronicsUnit.

Table 172 – Attributes of LTDSEquipmentProfile::PowerElectronicsUnit

| name | mult | type | description |
| --- | --- | --- | --- |
| maxP | 0..1 | [ActivePower](#UML33) | Maximum active power limit. This is the maximum (nameplate) limit for the unit. |
| minP | 0..1 | [ActivePower](#UML33) | Minimum active power limit. This is the minimum (nameplate) limit for the unit. |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 173 – Association ends of LTDSEquipmentProfile::PowerElectronicsUnit with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### PowerElectronicsWindUnit

Inheritance path = [PowerElectronicsUnit](#UML1978) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A wind generating unit that connects to the AC network with power electronics rather than rotating machines or an aggregation of such units.

Table 174 shows all attributes of PowerElectronicsWindUnit.

Table 174 – Attributes of LTDSEquipmentProfile::PowerElectronicsWindUnit

| name | mult | type | description |
| --- | --- | --- | --- |
| maxP | 0..1 | [ActivePower](#UML33) | inherited from: [PowerElectronicsUnit](#UML1978) |
| minP | 0..1 | [ActivePower](#UML33) | inherited from: [PowerElectronicsUnit](#UML1978) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 175 – Association ends of LTDSEquipmentProfile::PowerElectronicsWindUnit with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### (abstract) PowerSystemResource

Inheritance path = [IdentifiedObject](#UML12)

A power system resource (PSR) 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.

Table 176 shows all attributes of PowerSystemResource.

Table 176 – Attributes of LTDSEquipmentProfile::PowerSystemResource

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### PowerTransformer

Inheritance path = [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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 modelled 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.

Table 177 shows all attributes of PowerTransformer.

Table 177 – Attributes of LTDSEquipmentProfile::PowerTransformer

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 178 shows all association ends of PowerTransformer with other classes.

Table 178 – Association ends of LTDSEquipmentProfile::PowerTransformer with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### PowerTransformerEnd

Inheritance path = [TransformerEnd](#UML2012) : [IdentifiedObject](#UML12)

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 (TransformerEnd.endNumber=1) PowerTransformerEnd has non zero values on r, r0, x, and x0 while the low voltage (TransformerEnd.endNumber=2) PowerTransformerEnd has zero values for r, r0, x, and x0. Parameters are always provided, even if the PowerTransformerEnds have the same rated voltage. In this case, the parameters are provided at the PowerTransformerEnd which has TransformerEnd.endNumber equal to 1.

2) for a three Terminal PowerTransformer the three PowerTransformerEnds represent a star equivalent with each leg in the star represented by r, r0, x, and x0 values.

3) For a three Terminal transformer each PowerTransformerEnd shall have g, g0, b and b0 values corresponding to the no load losses distributed on the three PowerTransformerEnds. The total no load loss shunt impedances may also be placed at one of the PowerTransformerEnds, preferably the end numbered 1, having the shunt values on end 1. This is the preferred way.

4) 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).

Table 179 shows all attributes of PowerTransformerEnd.

Table 179 – Attributes of LTDSEquipmentProfile::PowerTransformerEnd

| name | mult | type | description |
| --- | --- | --- | --- |
| b | 1..1 | [Susceptance](#UML53) | Magnetizing branch susceptance (B mag). The value can be positive or negative. |
| connectionKind | 0..1 | [WindingConnection](#UML32) | Kind of connection. |
| ratedS | 0..1 | [ApparentPower](#UML40) | 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](#UML41) | Magnetizing branch conductance. |
| ratedU | 1..1 | [Voltage](#UML54) | 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 than or equal to ratedU for the lower voltage sides.  The attribute shall be a positive value. |
| r | 1..1 | [Resistance](#UML50) | Resistance (star-model) of the transformer end.  The attribute shall be equal to or greater than zero for non-equivalent transformers. |
| x | 1..1 | [Reactance](#UML47) | Positive sequence series reactance (star-model) of the transformer end. |
| endNumber | 1..1 | [Integer](#UML65) | inherited from: [TransformerEnd](#UML2012) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 180 shows all association ends of PowerTransformerEnd with other classes.

Table 180 – Association ends of LTDSEquipmentProfile::PowerTransformerEnd with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | PowerTransformer | 1..1 | [PowerTransformer](#UML1941) | The power transformer of this power transformer end. |
| 0..\* | BaseVoltage | 1..1 | [BaseVoltage](#UML1883) | inherited from: [TransformerEnd](#UML2012) |
| 0..\* | Terminal | 1..1 | [Terminal](#UML1888) | inherited from: [TransformerEnd](#UML2012) |

### (abstract) ProtectedSwitch

Inheritance path = [Switch](#UML1943) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A ProtectedSwitch is a switching device that can be operated by ProtectionEquipment.

Table 181 shows all attributes of ProtectedSwitch.

Table 181 – Attributes of LTDSEquipmentProfile::ProtectedSwitch

| name | mult | type | description |
| --- | --- | --- | --- |
| normalOpen | 1..1 | [Boolean](#UML57) | inherited from: [Switch](#UML1943) |
| ratedCurrent | 0..1 | [CurrentFlow](#UML42) | inherited from: [Switch](#UML1943) |
| retained | 1..1 | [Boolean](#UML57) | inherited from: [Switch](#UML1943) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 182 shows all association ends of ProtectedSwitch with other classes.

Table 182 – Association ends of LTDSEquipmentProfile::ProtectedSwitch with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### RatioTapChanger

Inheritance path = [TapChanger](#UML1998) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A tap changer that changes the voltage ratio impacting the voltage magnitude but not the phase angle across the transformer.

Angle sign convention (general): Positive value indicates a positive phase shift from the winding where the tap is located to the other winding (for a two-winding transformer).

Table 183 shows all attributes of RatioTapChanger.

Table 183 – Attributes of LTDSEquipmentProfile::RatioTapChanger

| name | mult | type | description |
| --- | --- | --- | --- |
| stepVoltageIncrement | 1..1 | [PerCent](#UML46) | Tap step increment, in per cent of rated voltage of the power transformer end, per step position.  When the increment is negative, the voltage decreases when the tap step increases. |
| highStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| lowStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| ltcFlag | 1..1 | [Boolean](#UML57) | inherited from: [TapChanger](#UML1998) |
| neutralStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| neutralU | 1..1 | [Voltage](#UML54) | inherited from: [TapChanger](#UML1998) |
| normalStep | 1..1 | [Integer](#UML65) | inherited from: [TapChanger](#UML1998) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 184 shows all association ends of RatioTapChanger with other classes.

Table 184 – Association ends of LTDSEquipmentProfile::RatioTapChanger with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..1 | TransformerEnd | 1..1 | [TransformerEnd](#UML2012) | Transformer end to which this ratio tap changer belongs. |
| 0..\* | RatioTapChangerTable | 0..1 | [RatioTapChangerTable](#UML2007) | The tap ratio table for this ratio tap changer. |
| 1..\* | TapChangerControl | 0..1 | [TapChangerControl](#UML1996) | inherited from: [TapChanger](#UML1998) |

### RatioTapChangerTable

Inheritance path = [IdentifiedObject](#UML12)

Describes a curve for how the voltage magnitude and impedance varies with the tap step.

Table 185 shows all attributes of RatioTapChangerTable.

Table 185 – Attributes of LTDSEquipmentProfile::RatioTapChangerTable

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### RatioTapChangerTablePoint

Inheritance path = [TapChangerTablePoint](#UML15)

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

Table 186 shows all attributes of RatioTapChangerTablePoint.

Table 186 – Attributes of LTDSEquipmentProfile::RatioTapChangerTablePoint

| name | mult | type | description |
| --- | --- | --- | --- |
| b | 0..1 | [PerCent](#UML46) | inherited from: [TapChangerTablePoint](#UML15) |
| g | 0..1 | [PerCent](#UML46) | inherited from: [TapChangerTablePoint](#UML15) |
| r | 0..1 | [PerCent](#UML46) | inherited from: [TapChangerTablePoint](#UML15) |
| ratio | 0..1 | [Float](#UML64) | inherited from: [TapChangerTablePoint](#UML15) |
| step | 1..1 | [Integer](#UML65) | inherited from: [TapChangerTablePoint](#UML15) |
| x | 0..1 | [PerCent](#UML46) | inherited from: [TapChangerTablePoint](#UML15) |

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

Table 187 – Association ends of LTDSEquipmentProfile::RatioTapChangerTablePoint with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 1..\* | RatioTapChangerTable | 1..1 | [RatioTapChangerTable](#UML2007) | Table of this point. |

### ReactiveCapabilityCurve

Inheritance path = [Curve](#UML1869) : [IdentifiedObject](#UML12)

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.

Table 188 shows all attributes of ReactiveCapabilityCurve.

Table 188 – Attributes of LTDSEquipmentProfile::ReactiveCapabilityCurve

| name | mult | type | description |
| --- | --- | --- | --- |
| curveStyle | 1..1 | [CurveStyle](#UML18) | inherited from: [Curve](#UML1869) |
| xUnit | 1..1 | [UnitSymbol](#UML30) | inherited from: [Curve](#UML1869) |
| y1Unit | 1..1 | [UnitSymbol](#UML30) | inherited from: [Curve](#UML1869) |
| y2Unit | 0..1 | [UnitSymbol](#UML30) | inherited from: [Curve](#UML1869) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### (abstract) RegulatingCondEq

Inheritance path = [EnergyConnection](#UML1920) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A type of conducting equipment that can regulate a quantity (i.e. voltage or flow) at a specific point in the network.

Table 189 shows all attributes of RegulatingCondEq.

Table 189 – Attributes of LTDSEquipmentProfile::RegulatingCondEq

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 190 – Association ends of LTDSEquipmentProfile::RegulatingCondEq with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | RegulatingControl | 0..1 | [RegulatingControl](#UML1995) | The regulating control scheme in which this equipment participates. |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### RegulatingControl

Inheritance path = [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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.

The specified terminal shall be associated with the connectivity node of the controlled point. 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.

The attribute minAllowedTargetValue and maxAllowedTargetValue are required in the following cases:

- For a power generating module operated in power factor control mode to specify maximum and minimum power factor values;

- Whenever it is necessary to have an off center target voltage for the tap changer regulator. For instance, due to long cables to off shore wind farms and the need to have a simpler setup at the off shore transformer platform, the voltage is controlled from the land at the connection point for the off shore wind farm. Since there usually is a voltage rise along the cable, there is typical and overvoltage of up 3-4 kV compared to the on shore station. Thus in normal operation the tap changer on the on shore station is operated with a target set point, which is in the lower parts of the dead band.

The attributes minAllowedTargetValue and maxAllowedTargetValue are not related to the attribute targetDeadband and thus they are not treated as an alternative of the targetDeadband. They are needed due to limitations in the local substation controller. The attribute targetDeadband is used to prevent the power flow from move the tap position in circles (hunting) that is to be used regardless of the attributes minAllowedTargetValue and maxAllowedTargetValue.

Table 191 shows all attributes of RegulatingControl.

Table 191 – Attributes of LTDSEquipmentProfile::RegulatingControl

| name | mult | type | description |
| --- | --- | --- | --- |
| mode | 1..1 | [RegulatingControlModeKind](#UML27) | 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](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 192 – Association ends of LTDSEquipmentProfile::RegulatingControl with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Terminal | 1..1 | [Terminal](#UML1888) | The terminal associated with this regulating control. The terminal is associated instead of a node, since the terminal could connect into either a topological node or a connectivity node. Sometimes it is useful to model regulation at a terminal of a bus bar object. |

### RegularTimePoint root class

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

Table 193 shows all attributes of RegularTimePoint.

Table 193 – Attributes of LTDSEquipmentProfile::RegularTimePoint

| name | mult | type | description |
| --- | --- | --- | --- |
| sequenceNumber | 1..1 | [Integer](#UML65) | 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. To specify values for the start time, use sequence number 0. The sequence number cannot be negative. |
| value1 | 1..1 | [Float](#UML64) | The first value at the time. The meaning of the value is defined by the derived type of the associated schedule. |
| value2 | 0..1 | [Float](#UML64) | The second value at the time. The meaning of the value is defined by the derived type of the associated schedule. |

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

Table 194 – Association ends of LTDSEquipmentProfile::RegularTimePoint with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 1..\* | IntervalSchedule | 1..1 | [RegularIntervalSchedule](#UML1876) | Regular interval schedule containing this time point. |

### (abstract) RegularIntervalSchedule

Inheritance path = [BasicIntervalSchedule](#UML1875) : [IdentifiedObject](#UML12)

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

Table 195 shows all attributes of RegularIntervalSchedule.

Table 195 – Attributes of LTDSEquipmentProfile::RegularIntervalSchedule

| name | mult | type | description |
| --- | --- | --- | --- |
| timeStep | 1..1 | [Seconds](#UML52) | The time between each pair of subsequent regular time points in sequence order. |
| endTime | 1..1 | [DateTime](#UML63) | The time for the last time point. The value can be a time of day, not a specific date. |
| startTime | 1..1 | [DateTime](#UML63) | inherited from: [BasicIntervalSchedule](#UML1875) |
| value1Unit | 1..1 | [UnitSymbol](#UML30) | inherited from: [BasicIntervalSchedule](#UML1875) |
| value2Unit | 0..1 | [UnitSymbol](#UML30) | inherited from: [BasicIntervalSchedule](#UML1875) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### ReportingGroup

Inheritance path = [IdentifiedObject](#UML12)

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

Table 196 shows all attributes of ReportingGroup.

Table 196 – Attributes of LTDSEquipmentProfile::ReportingGroup

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### (abstract) RotatingMachine

Inheritance path = [RegulatingCondEq](#UML1926) : [EnergyConnection](#UML1920) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A rotating machine which may be used as a generator or motor.

Table 197 shows all attributes of RotatingMachine.

Table 197 – Attributes of LTDSEquipmentProfile::RotatingMachine

| name | mult | type | description |
| --- | --- | --- | --- |
| ratedPowerFactor | 0..1 | [Float](#UML64) | Power factor (nameplate data). It is primarily used for short circuit data exchange according to IEC 60909. The attribute cannot be a negative value. |
| ratedS | 0..1 | [ApparentPower](#UML40) | Nameplate apparent power rating for the unit.  The attribute shall have a positive value. |
| ratedU | 0..1 | [Voltage](#UML54) | Rated voltage (nameplate data, Ur in IEC 60909-0). It is primarily used for short circuit data exchange according to IEC 60909.  The attribute shall be a positive value. |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 198 – Association ends of LTDSEquipmentProfile::RotatingMachine with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 1..\* | GeneratingUnit | 0..1 | [GeneratingUnit](#UML1971) | A synchronous machine may operate as a generator and as such becomes a member of a generating unit. |
| 0..\* | RegulatingControl | 0..1 | [RegulatingControl](#UML1995) | inherited from: [RegulatingCondEq](#UML1926) |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### Season

Inheritance path = [IdentifiedObject](#UML12)

A specified time period of the year.

Table 199 shows all attributes of Season.

Table 199 – Attributes of LTDSEquipmentProfile::Season

| name | mult | type | description |
| --- | --- | --- | --- |
| endDate | 1..1 | [MonthDay](#UML66) | Date season ends. |
| startDate | 1..1 | [MonthDay](#UML66) | Date season starts. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### (abstract) Sensor

Inheritance path = [AuxiliaryEquipment](#UML1953) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

This class describe devices that transform a measured quantity into signals that can be presented at displays, used in control or be recorded.

Table 200 shows all attributes of Sensor.

Table 200 – Attributes of LTDSEquipmentProfile::Sensor

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 201 shows all association ends of Sensor with other classes.

Table 201 – Association ends of LTDSEquipmentProfile::Sensor with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Terminal | 1..1 | [Terminal](#UML1888) | inherited from: [AuxiliaryEquipment](#UML1953) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### (abstract) SeasonDayTypeSchedule

Inheritance path = [RegularIntervalSchedule](#UML1876) : [BasicIntervalSchedule](#UML1875) : [IdentifiedObject](#UML12)

A time schedule covering a 24 hour period, with curve data for a specific type of season and day.

Table 202 shows all attributes of SeasonDayTypeSchedule.

Table 202 – Attributes of LTDSEquipmentProfile::SeasonDayTypeSchedule

| name | mult | type | description |
| --- | --- | --- | --- |
| timeStep | 1..1 | [Seconds](#UML52) | inherited from: [RegularIntervalSchedule](#UML1876) |
| endTime | 1..1 | [DateTime](#UML63) | inherited from: [RegularIntervalSchedule](#UML1876) |
| startTime | 1..1 | [DateTime](#UML63) | inherited from: [BasicIntervalSchedule](#UML1875) |
| value1Unit | 1..1 | [UnitSymbol](#UML30) | inherited from: [BasicIntervalSchedule](#UML1875) |
| value2Unit | 0..1 | [UnitSymbol](#UML30) | inherited from: [BasicIntervalSchedule](#UML1875) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 203 shows all association ends of SeasonDayTypeSchedule with other classes.

Table 203 – Association ends of LTDSEquipmentProfile::SeasonDayTypeSchedule with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | DayType | 1..1 | [DayType](#UML1868) | DayType for the Schedule. |
| 0..\* | Season | 1..1 | [Season](#UML2009) | Season for the Schedule. |

### SeriesCompensator

Inheritance path = [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A Series Compensator is a series capacitor or reactor or an AC transmission line without charging susceptance. It is a two terminal device.

Table 204 shows all attributes of SeriesCompensator.

Table 204 – Attributes of LTDSEquipmentProfile::SeriesCompensator

| name | mult | type | description |
| --- | --- | --- | --- |
| r | 1..1 | [Resistance](#UML50) | Positive sequence resistance. |
| x | 1..1 | [Reactance](#UML47) | Positive sequence reactance. |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 205 shows all association ends of SeriesCompensator with other classes.

Table 205 – Association ends of LTDSEquipmentProfile::SeriesCompensator with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### (abstract) ShuntCompensator

Inheritance path = [RegulatingCondEq](#UML1926) : [EnergyConnection](#UML1920) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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 bPerSection indicates that the compensator is a reactor. ShuntCompensator is a single terminal device. Ground is implied.

Table 206 shows all attributes of ShuntCompensator.

Table 206 – Attributes of LTDSEquipmentProfile::ShuntCompensator

| name | mult | type | description |
| --- | --- | --- | --- |
| aVRDelay | 0..1 | [Seconds](#UML52) | An automatic voltage regulation delay (AVRDelay) which is the time delay from a change in voltage to when the capacitor is allowed to change state. This filters out temporary changes in voltage. |
| grounded | 0..1 | [Boolean](#UML57) | Used for Yn and Zn connections. True if the neutral is solidly grounded. |
| maximumSections | 1..1 | [Integer](#UML65) | The maximum number of sections that may be switched in. |
| nomU | 1..1 | [Voltage](#UML54) | 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](#UML65) | The normal number of sections switched in. The value shall be between zero and ShuntCompensator.maximumSections. |
| voltageSensitivity | 0..1 | [VoltagePerReactivePower](#UML55) | Voltage sensitivity required for the device to regulate the bus voltage, in voltage/reactive power. |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 207 shows all association ends of ShuntCompensator with other classes.

Table 207 – Association ends of LTDSEquipmentProfile::ShuntCompensator with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | RegulatingControl | 0..1 | [RegulatingControl](#UML1995) | inherited from: [RegulatingCondEq](#UML1926) |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### SolarGeneratingUnit

Inheritance path = [GeneratingUnit](#UML1971) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A solar thermal generating unit, connected to the grid by means of a rotating machine. This class does not represent photovoltaic (PV) generation.

Table 208 shows all attributes of SolarGeneratingUnit.

Table 208 – Attributes of LTDSEquipmentProfile::SolarGeneratingUnit

| name | mult | type | description |
| --- | --- | --- | --- |
| genControlSource | 0..1 | [GeneratorControlSource](#UML20) | inherited from: [GeneratingUnit](#UML1971) |
| governorSCD | 0..1 | [PerCent](#UML46) | inherited from: [GeneratingUnit](#UML1971) |
| longPF | 0..1 | [Float](#UML64) | inherited from: [GeneratingUnit](#UML1971) |
| maximumAllowableSpinningReserve | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| maxOperatingP | 1..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| minOperatingP | 1..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| nominalP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| ratedGrossMaxP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| ratedGrossMinP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| ratedNetMaxP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| shortPF | 0..1 | [Float](#UML64) | inherited from: [GeneratingUnit](#UML1971) |
| startupCost | 0..1 | [Money](#UML45) | inherited from: [GeneratingUnit](#UML1971) |
| variableCost | 0..1 | [Money](#UML45) | inherited from: [GeneratingUnit](#UML1971) |
| startupTime | 0..1 | [Seconds](#UML52) | inherited from: [GeneratingUnit](#UML1971) |
| totalEfficiency | 0..1 | [PerCent](#UML46) | inherited from: [GeneratingUnit](#UML1971) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 209 shows all association ends of SolarGeneratingUnit with other classes.

Table 209 – Association ends of LTDSEquipmentProfile::SolarGeneratingUnit with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | SolarPowerPlant | 0..1 | [SolarPowerPlant](#UML1997) | (European) A solar power plant may have solar generating units. |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### (European) SolarPowerPlant

Inheritance path = [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

Solar power plant.

Table 210 shows all attributes of SolarPowerPlant.

Table 210 – Attributes of LTDSEquipmentProfile::SolarPowerPlant

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### StaticVarCompensator

Inheritance path = [RegulatingCondEq](#UML1926) : [EnergyConnection](#UML1920) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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.

Table 211 shows all attributes of StaticVarCompensator.

Table 211 – Attributes of LTDSEquipmentProfile::StaticVarCompensator

| name | mult | type | description |
| --- | --- | --- | --- |
| capacitiveRating | 1..1 | [Reactance](#UML47) | Capacitive reactance at maximum capacitive reactive power. Shall always be positive. |
| inductiveRating | 1..1 | [Reactance](#UML47) | Inductive reactance at maximum inductive reactive power. Shall always be negative. |
| slope | 1..1 | [VoltagePerReactivePower](#UML55) | 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.  The attribute shall be a positive value or zero. |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 212 – Association ends of LTDSEquipmentProfile::StaticVarCompensator with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | RegulatingControl | 0..1 | [RegulatingControl](#UML1995) | inherited from: [RegulatingCondEq](#UML1926) |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### StationSupply

Inheritance path = [EnergyConsumer](#UML1921) : [EnergyConnection](#UML1920) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

Station supply with load derived from the station output.

Table 213 shows all attributes of StationSupply.

Table 213 – Attributes of LTDSEquipmentProfile::StationSupply

| name | mult | type | description |
| --- | --- | --- | --- |
| pfixed | 0..1 | [ActivePower](#UML33) | inherited from: [EnergyConsumer](#UML1921) |
| pfixedPct | 0..1 | [PerCent](#UML46) | inherited from: [EnergyConsumer](#UML1921) |
| qfixed | 0..1 | [ReactivePower](#UML48) | inherited from: [EnergyConsumer](#UML1921) |
| qfixedPct | 0..1 | [PerCent](#UML46) | inherited from: [EnergyConsumer](#UML1921) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 214 – Association ends of LTDSEquipmentProfile::StationSupply with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | LoadResponse | 0..1 | [LoadResponseCharacteristic](#UML1893) | inherited from: [EnergyConsumer](#UML1921) |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### SubGeographicalRegion

Inheritance path = [IdentifiedObject](#UML12)

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

Table 215 shows all attributes of SubGeographicalRegion.

Table 215 – Attributes of LTDSEquipmentProfile::SubGeographicalRegion

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 216 – Association ends of LTDSEquipmentProfile::SubGeographicalRegion with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Region | 1..1 | [GeographicalRegion](#UML1862) | The geographical region which this sub-geographical region is within. |

### SubLoadArea

Inheritance path = [EnergyArea](#UML1865) : [IdentifiedObject](#UML12)

The class is the second level in a hierarchical structure for grouping of loads for the purpose of load flow load scaling.

Table 217 shows all attributes of SubLoadArea.

Table 217 – Attributes of LTDSEquipmentProfile::SubLoadArea

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 218 – Association ends of LTDSEquipmentProfile::SubLoadArea with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 1..\* | LoadArea | 1..1 | [LoadArea](#UML1866) | The LoadArea where the SubLoadArea belongs. |

### Substation

Inheritance path = [EquipmentContainer](#UML1985) : [ConnectivityNodeContainer](#UML1984) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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.

Table 219 shows all attributes of Substation.

Table 219 – Attributes of LTDSEquipmentProfile::Substation

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 220 – Association ends of LTDSEquipmentProfile::Substation with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Region | 1..1 | [SubGeographicalRegion](#UML2010) | The SubGeographicalRegion containing the substation. |

### SurgeArrester

Inheritance path = [AuxiliaryEquipment](#UML1953) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

Shunt device, installed on the network, usually in the proximity of electrical equipment in order to protect the said equipment against transient voltage transients caused by lightning or switching activity.

Table 221 shows all attributes of SurgeArrester.

Table 221 – Attributes of LTDSEquipmentProfile::SurgeArrester

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 222 – Association ends of LTDSEquipmentProfile::SurgeArrester with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Terminal | 1..1 | [Terminal](#UML1888) | inherited from: [AuxiliaryEquipment](#UML1953) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### Switch

Inheritance path = [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A generic device designed to close, or open, or both, one or more electric circuits. All switches are two terminal devices including grounding switches. The ACDCTerminal.connected at the two sides of the switch shall not be considered for assessing switch connectivity, i.e. only Switch.open, .normalOpen and .locked are relevant.

Table 223 shows all attributes of Switch.

Table 223 – Attributes of LTDSEquipmentProfile::Switch

| name | mult | type | description |
| --- | --- | --- | --- |
| normalOpen | 1..1 | [Boolean](#UML57) | 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](#UML42) | The maximum continuous current carrying capacity in amps governed by the device material and construction.  The attribute shall be a positive value. |
| retained | 1..1 | [Boolean](#UML57) | Branch is retained in the topological solution. The flow through retained switches will normally be calculated in power flow. |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 224 – Association ends of LTDSEquipmentProfile::Switch with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### SwitchSchedule

Inheritance path = [SeasonDayTypeSchedule](#UML1877) : [RegularIntervalSchedule](#UML1876) : [BasicIntervalSchedule](#UML1875) : [IdentifiedObject](#UML12)

A schedule of switch positions. If RegularTimePoint.value1 is 0, the switch is open. If 1, the switch is closed.

Table 225 shows all attributes of SwitchSchedule.

Table 225 – Attributes of LTDSEquipmentProfile::SwitchSchedule

| name | mult | type | description |
| --- | --- | --- | --- |
| timeStep | 1..1 | [Seconds](#UML52) | inherited from: [RegularIntervalSchedule](#UML1876) |
| endTime | 1..1 | [DateTime](#UML63) | inherited from: [RegularIntervalSchedule](#UML1876) |
| startTime | 1..1 | [DateTime](#UML63) | inherited from: [BasicIntervalSchedule](#UML1875) |
| value1Unit | 1..1 | [UnitSymbol](#UML30) | inherited from: [BasicIntervalSchedule](#UML1875) |
| value2Unit | 0..1 | [UnitSymbol](#UML30) | inherited from: [BasicIntervalSchedule](#UML1875) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 226 – Association ends of LTDSEquipmentProfile::SwitchSchedule with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Switch | 1..1 | [Switch](#UML1943) | A SwitchSchedule is associated with a Switch. |
| 0..\* | DayType | 1..1 | [DayType](#UML1868) | inherited from: [SeasonDayTypeSchedule](#UML1877) |
| 0..\* | Season | 1..1 | [Season](#UML2009) | inherited from: [SeasonDayTypeSchedule](#UML1877) |

### SynchronousMachine

Inheritance path = [RotatingMachine](#UML1929) : [RegulatingCondEq](#UML1926) : [EnergyConnection](#UML1920) : [ConductingEquipment](#UML1907) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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.

Table 227 shows all attributes of SynchronousMachine.

Table 227 – Attributes of LTDSEquipmentProfile::SynchronousMachine

| name | mult | type | description |
| --- | --- | --- | --- |
| maxQ | 0..1 | [ReactivePower](#UML48) | Maximum reactive power limit. This is the maximum (nameplate) limit for the unit. |
| minQ | 0..1 | [ReactivePower](#UML48) | Minimum reactive power limit for the unit. |
| qPercent | 0..1 | [PerCent](#UML46) | Part of the coordinated reactive control that comes from this machine. The attribute is used as a participation factor not necessarily summing up to 100% for the participating devices in the control. |
| type | 1..1 | [SynchronousMachineKind](#UML28) | Modes that this synchronous machine can operate in. |
| ratedPowerFactor | 0..1 | [Float](#UML64) | inherited from: [RotatingMachine](#UML1929) |
| ratedS | 0..1 | [ApparentPower](#UML40) | inherited from: [RotatingMachine](#UML1929) |
| ratedU | 0..1 | [Voltage](#UML54) | inherited from: [RotatingMachine](#UML1929) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 228 – Association ends of LTDSEquipmentProfile::SynchronousMachine with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 1..\* | InitialReactiveCapabilityCurve | 0..1 | [ReactiveCapabilityCurve](#UML1872) | The default reactive capability curve for use by a synchronous machine. |
| 1..\* | GeneratingUnit | 0..1 | [GeneratingUnit](#UML1971) | inherited from: [RotatingMachine](#UML1929) |
| 0..\* | RegulatingControl | 0..1 | [RegulatingControl](#UML1995) | inherited from: [RegulatingCondEq](#UML1926) |
| 0..\* | BaseVoltage | 0..1 | [BaseVoltage](#UML1883) | inherited from: [ConductingEquipment](#UML1907) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### (abstract) TapChanger

Inheritance path = [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

Mechanism for changing transformer winding tap positions.

Table 229 shows all attributes of TapChanger.

Table 229 – Attributes of LTDSEquipmentProfile::TapChanger

| name | mult | type | description |
| --- | --- | --- | --- |
| highStep | 1..1 | [Integer](#UML65) | Highest possible tap step position, advance from neutral.  The attribute shall be greater than lowStep. |
| lowStep | 1..1 | [Integer](#UML65) | Lowest possible tap step position, retard from neutral. |
| ltcFlag | 1..1 | [Boolean](#UML57) | Specifies whether or not a TapChanger has load tap changing capabilities. |
| neutralStep | 1..1 | [Integer](#UML65) | The neutral tap step position for this winding.  The attribute shall be equal to or greater than lowStep and equal or less than highStep.  It is the step position where the voltage is neutralU when the other terminals of the transformer are at the ratedU. If there are other tap changers on the transformer those taps are kept constant at their neutralStep. |
| neutralU | 1..1 | [Voltage](#UML54) | Voltage at which the winding operates at the neutral tap setting. It is the voltage at the terminal of the PowerTransformerEnd associated with the tap changer when all tap changers on the transformer are at their neutralStep position. Normally neutralU of the tap changer is the same as ratedU of the PowerTransformerEnd, but it can differ in special cases such as when the tapping mechanism is separate from the winding more common on lower voltage transformers.  This attribute is not relevant for PhaseTapChangerAsymmetrical, PhaseTapChangerSymmetrical and PhaseTapChangerLinear. |
| normalStep | 1..1 | [Integer](#UML65) | 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 to or greater than lowStep and equal to or less than highStep. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 230 – Association ends of LTDSEquipmentProfile::TapChanger with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 1..\* | TapChangerControl | 0..1 | [TapChangerControl](#UML1996) | The regulating control scheme in which this tap changer participates. |

### TapChangerControl

Inheritance path = [RegulatingControl](#UML1995) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

Describes behaviour 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.

Table 231 shows all attributes of TapChangerControl.

Table 231 – Attributes of LTDSEquipmentProfile::TapChangerControl

| name | mult | type | description |
| --- | --- | --- | --- |
| mode | 1..1 | [RegulatingControlModeKind](#UML27) | inherited from: [RegulatingControl](#UML1995) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

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

Table 232 – Association ends of LTDSEquipmentProfile::TapChangerControl with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Terminal | 1..1 | [Terminal](#UML1888) | inherited from: [RegulatingControl](#UML1995) |

### (abstract) TapChangerTablePoint root class

Describes each tap step in the tabular curve.

Table 233 shows all attributes of TapChangerTablePoint.

Table 233 – Attributes of LTDSEquipmentProfile::TapChangerTablePoint

| name | mult | type | description |
| --- | --- | --- | --- |
| b | 0..1 | [PerCent](#UML46) | The magnetizing branch susceptance deviation as a percentage 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](#UML46) | The magnetizing branch conductance deviation as a percentage 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](#UML46) | The resistance deviation as a percentage 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 | [Float](#UML64) | The voltage at the tap step divided by rated voltage of the transformer end having the tap changer. Hence this is a value close to one.  For example, if the ratio at step 1 is 1.01, and the rated voltage of the transformer end is 110kV, then the voltage obtained by setting the tap changer to step 1 to is 111.1kV. |
| step | 1..1 | [Integer](#UML65) | The tap step. |
| x | 0..1 | [PerCent](#UML46) | The series reactance deviation as a percentage 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. |

### Terminal

Inheritance path = [ACDCTerminal](#UML1884) : [IdentifiedObject](#UML12)

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

Table 234 shows all attributes of Terminal.

Table 234 – Attributes of LTDSEquipmentProfile::Terminal

| name | mult | type | description |
| --- | --- | --- | --- |
| phases | 0..1 | [PhaseCode](#UML26) | Represents the normal network phasing condition. If the attribute is missing, three phases (ABC) shall be assumed, except for terminals of grounding classes (specializations of EarthFaultCompensator, GroundDisconnector, and Ground) which will be assumed to be N. Therefore, phase code ABCN is explicitly declared when needed, e.g. for star point grounding equipment.  The phase code on terminals connecting same ConnectivityNode or same TopologicalNode as well as for equipment between two terminals shall be consistent. |
| sequenceNumber | 1..1 | [Integer](#UML65) | inherited from: [ACDCTerminal](#UML1884) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 235 shows all association ends of Terminal with other classes.

Table 235 – Association ends of LTDSEquipmentProfile::Terminal with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | ConductingEquipment | 1..1 | [ConductingEquipment](#UML1907) | 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. |
| 0..\* | ConnectivityNode | 0..1 | [ConnectivityNode](#UML1874) | The connectivity node to which this terminal connects with zero impedance. |
| 1..\* | BusNameMarker | 0..1 | [BusNameMarker](#UML1864) | inherited from: [ACDCTerminal](#UML1884) |

### ThermalGeneratingUnit

Inheritance path = [GeneratingUnit](#UML1971) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 236 shows all attributes of ThermalGeneratingUnit.

Table 236 – Attributes of LTDSEquipmentProfile::ThermalGeneratingUnit

| name | mult | type | description |
| --- | --- | --- | --- |
| genControlSource | 0..1 | [GeneratorControlSource](#UML20) | inherited from: [GeneratingUnit](#UML1971) |
| governorSCD | 0..1 | [PerCent](#UML46) | inherited from: [GeneratingUnit](#UML1971) |
| longPF | 0..1 | [Float](#UML64) | inherited from: [GeneratingUnit](#UML1971) |
| maximumAllowableSpinningReserve | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| maxOperatingP | 1..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| minOperatingP | 1..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| nominalP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| ratedGrossMaxP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| ratedGrossMinP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| ratedNetMaxP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| shortPF | 0..1 | [Float](#UML64) | inherited from: [GeneratingUnit](#UML1971) |
| startupCost | 0..1 | [Money](#UML45) | inherited from: [GeneratingUnit](#UML1971) |
| variableCost | 0..1 | [Money](#UML45) | inherited from: [GeneratingUnit](#UML1971) |
| startupTime | 0..1 | [Seconds](#UML52) | inherited from: [GeneratingUnit](#UML1971) |
| totalEfficiency | 0..1 | [PerCent](#UML46) | inherited from: [GeneratingUnit](#UML1971) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 237 shows all association ends of ThermalGeneratingUnit with other classes.

Table 237 – Association ends of LTDSEquipmentProfile::ThermalGeneratingUnit with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..1 | CAESPlant | 0..1 | [CAESPlant](#UML1904) | A thermal generating unit may be a member of a compressed air energy storage plant. |
| 0..\* | CogenerationPlant | 0..1 | [CogenerationPlant](#UML1982) | A thermal generating unit may be a member of a cogeneration plant. |
| 0..\* | CombinedCyclePlant | 0..1 | [CombinedCyclePlant](#UML1994) | A thermal generating unit may be a member of a combined cycle plant. |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### TieFlow

Inheritance path = [IdentifiedObject](#UML12)

Defines the structure (in terms of location and direction) of the net interchange constraint for a control area. This constraint may be used by either AGC or power flow.

Table 238 shows all attributes of TieFlow.

Table 238 – Attributes of LTDSEquipmentProfile::TieFlow

| name | mult | type | description |
| --- | --- | --- | --- |
| positiveFlowIn | 1..1 | [Boolean](#UML57) | Specifies the sign of the tie flow associated with a control area. True if positive flow into the terminal (load convention) is also positive flow into the control area. See the description of ControlArea for further explanation of how TieFlow.positiveFlowIn is used. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 239 shows all association ends of TieFlow with other classes.

Table 239 – Association ends of LTDSEquipmentProfile::TieFlow with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | ControlArea | 1..1 | [ControlArea](#UML1983) | The control area of the tie flows. |
| 0..2 | Terminal | 1..1 | [Terminal](#UML1888) | The terminal to which this tie flow belongs. |

### (abstract) TransformerEnd

Inheritance path = [IdentifiedObject](#UML12)

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.

Table 240 shows all attributes of TransformerEnd.

Table 240 – Attributes of LTDSEquipmentProfile::TransformerEnd

| name | mult | type | description |
| --- | --- | --- | --- |
| endNumber | 1..1 | [Integer](#UML65) | 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. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 241 shows all association ends of TransformerEnd with other classes.

Table 241 – Association ends of LTDSEquipmentProfile::TransformerEnd with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 1..1 | [BaseVoltage](#UML1883) | Base voltage of the transformer end. This is essential for PU calculation. |
| 0..\* | Terminal | 1..1 | [Terminal](#UML1888) | Terminal of the power transformer to which this transformer end belongs. |

### VoltageLevel

Inheritance path = [EquipmentContainer](#UML1985) : [ConnectivityNodeContainer](#UML1984) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

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

Table 242 shows all attributes of VoltageLevel.

Table 242 – Attributes of LTDSEquipmentProfile::VoltageLevel

| name | mult | type | description |
| --- | --- | --- | --- |
| highVoltageLimit | 0..1 | [Voltage](#UML54) | The bus bar's high voltage limit.  The limit applies to all equipment and nodes contained in a given VoltageLevel. It is not required that it is exchanged in pair with lowVoltageLimit. It is preferable to use operational VoltageLimit, which prevails, if present. |
| lowVoltageLimit | 0..1 | [Voltage](#UML54) | The bus bar's low voltage limit.  The limit applies to all equipment and nodes contained in a given VoltageLevel. It is not required that it is exchanged in pair with highVoltageLimit. It is preferable to use operational VoltageLimit, which prevails, if present. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 243 shows all association ends of VoltageLevel with other classes.

Table 243 – Association ends of LTDSEquipmentProfile::VoltageLevel with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | BaseVoltage | 1..1 | [BaseVoltage](#UML1883) | The base voltage used for all equipment within the voltage level. |
| 0..\* | Substation | 1..1 | [Substation](#UML1991) | The substation of the voltage level. |

### VoltageLimit

Inheritance path = [OperationalLimit](#UML1894) : [IdentifiedObject](#UML12)

Operational limit applied to voltage.

The use of operational VoltageLimit is preferred instead of limits defined at VoltageLevel. The operational VoltageLimits are used, if present.

Table 244 shows all attributes of VoltageLimit.

Table 244 – Attributes of LTDSEquipmentProfile::VoltageLimit

| name | mult | type | description |
| --- | --- | --- | --- |
| normalValue | 1..1 | [Voltage](#UML54) | The normal limit on voltage. High or low limit nature of the limit depends upon the properties of the operational limit type. The attribute shall be a positive value or zero. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 245 shows all association ends of VoltageLimit with other classes.

Table 245 – Association ends of LTDSEquipmentProfile::VoltageLimit with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | OperationalLimitType | 1..1 | [OperationalLimitType](#UML1900) | inherited from: [OperationalLimit](#UML1894) |
| 0..\* | OperationalLimitSet | 1..1 | [OperationalLimitSet](#UML1899) | inherited from: [OperationalLimit](#UML1894) |

### WaveTrap

Inheritance path = [AuxiliaryEquipment](#UML1953) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

Line traps are devices that impede high frequency power line carrier signals yet present a negligible impedance at the main power frequency.

Table 246 shows all attributes of WaveTrap.

Table 246 – Attributes of LTDSEquipmentProfile::WaveTrap

| name | mult | type | description |
| --- | --- | --- | --- |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 247 shows all association ends of WaveTrap with other classes.

Table 247 – Association ends of LTDSEquipmentProfile::WaveTrap with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Terminal | 1..1 | [Terminal](#UML1888) | inherited from: [AuxiliaryEquipment](#UML1953) |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### WindGeneratingUnit

Inheritance path = [GeneratingUnit](#UML1971) : [Equipment](#UML1906) : [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

A wind driven generating unit, connected to the grid by means of a rotating machine. May be used to represent a single turbine or an aggregation.

Table 248 shows all attributes of WindGeneratingUnit.

Table 248 – Attributes of LTDSEquipmentProfile::WindGeneratingUnit

| name | mult | type | description |
| --- | --- | --- | --- |
| windGenUnitType | 1..1 | [WindGenUnitKind](#UML31) | The kind of wind generating unit. |
| genControlSource | 0..1 | [GeneratorControlSource](#UML20) | inherited from: [GeneratingUnit](#UML1971) |
| governorSCD | 0..1 | [PerCent](#UML46) | inherited from: [GeneratingUnit](#UML1971) |
| longPF | 0..1 | [Float](#UML64) | inherited from: [GeneratingUnit](#UML1971) |
| maximumAllowableSpinningReserve | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| maxOperatingP | 1..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| minOperatingP | 1..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| nominalP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| ratedGrossMaxP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| ratedGrossMinP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| ratedNetMaxP | 0..1 | [ActivePower](#UML33) | inherited from: [GeneratingUnit](#UML1971) |
| shortPF | 0..1 | [Float](#UML64) | inherited from: [GeneratingUnit](#UML1971) |
| startupCost | 0..1 | [Money](#UML45) | inherited from: [GeneratingUnit](#UML1971) |
| variableCost | 0..1 | [Money](#UML45) | inherited from: [GeneratingUnit](#UML1971) |
| startupTime | 0..1 | [Seconds](#UML52) | inherited from: [GeneratingUnit](#UML1971) |
| totalEfficiency | 0..1 | [PerCent](#UML46) | inherited from: [GeneratingUnit](#UML1971) |
| aggregate | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| normallyInService | 0..1 | [Boolean](#UML57) | inherited from: [Equipment](#UML1906) |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

Table 249 shows all association ends of WindGeneratingUnit with other classes.

Table 249 – Association ends of LTDSEquipmentProfile::WindGeneratingUnit with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | WindPowerPlant | 0..1 | [WindPowerPlant](#UML2006) | (European) A wind power plant may have wind generating units. |
| 0..\* | EquipmentContainer | 0..1 | [EquipmentContainer](#UML1985) | inherited from: [Equipment](#UML1906) |

### (European) WindPowerPlant

Inheritance path = [PowerSystemResource](#UML1902) : [IdentifiedObject](#UML12)

Wind power plant.

Table 250 shows all attributes of WindPowerPlant.

Table 250 – Attributes of LTDSEquipmentProfile::WindPowerPlant

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

### ControlAreaTypeKind enumeration

The type of control area.

Table 251 shows all literals of ControlAreaTypeKind.

Table 251 – Literals of LTDSEquipmentProfile::ControlAreaTypeKind

| literal | value | description |
| --- | --- | --- |
| AGC |  | Used for automatic generation control. |
| Forecast |  | Used for load forecast. |
| Interchange |  | Used for interchange specification or control. |

### Currency enumeration

Monetary currencies. ISO 4217 standard including 3-character currency code.

Table 252 shows all literals of Currency.

Table 252 – Literals of LTDSEquipmentProfile::Currency

| literal | value | description |
| --- | --- | --- |
| AED | 784 | United Arab Emirates dirham. |
| AFN | 971 | Afghan afghani. |
| ALL | 008 | Albanian lek. |
| AMD | 051 | Armenian dram. |
| ANG | 532 | Netherlands Antillean guilder. |
| AOA | 973 | Angolan kwanza. |
| ARS | 032 | Argentine peso. |
| AUD | 036 | Australian dollar. |
| AWG | 533 | Aruban florin. |
| AZN | 944 | Azerbaijani manat. |
| BAM | 977 | Bosnia and Herzegovina convertible mark. |
| BBD | 052 | Barbados dollar. |
| BDT | 050 | Bangladeshi taka. |
| BGN | 975 | Bulgarian lev. |
| BHD | 048 | Bahraini dinar. |
| BIF | 108 | Burundian franc. |
| BMD | 060 | Bermudian dollar (customarily known as Bermuda dollar). |
| BND | 096 | Brunei dollar. |
| BOB | 068 | Boliviano. |
| BOV | 984 | Bolivian Mvdol (funds code). |
| BRL | 986 | Brazilian real. |
| BSD | 044 | Bahamian dollar. |
| BTN | 064 | Bhutanese ngultrum. |
| BWP | 072 | Botswana pula. |
| BYR | 974 | Belarusian ruble. |
| BZD | 084 | Belize dollar. |
| CAD | 124 | Canadian dollar. |
| CDF | 976 | Congolese franc. |
| CHF | 756 | Swiss franc. |
| CLF | 990 | Unidad de Fomento (funds code), Chile. |
| CLP | 152 | Chilean peso. |
| CNY | 156 | Chinese yuan. |
| COP | 170 | Colombian peso. |
| COU | 970 | Unidad de Valor Real. |
| CRC | 188 | Costa Rican colon. |
| CUC | 931 | Cuban convertible peso. |
| CUP | 192 | Cuban peso. |
| CVE | 132 | Cape Verde escudo. |
| CZK | 203 | Czech koruna. |
| DJF | 262 | Djiboutian franc. |
| DKK | 208 | Danish krone. |
| DOP | 214 | Dominican peso. |
| DZD | 012 | Algerian dinar. |
| EEK | 233 | Estonian kroon. |
| EGP | 818 | Egyptian pound. |
| ERN | 232 | Eritrean nakfa. |
| ETB | 230 | Ethiopian birr. |
| EUR | 978 | Euro. |
| FJD | 242 | Fiji dollar. |
| FKP | 238 | Falkland Islands pound. |
| GBP | 826 | Pound sterling. |
| GEL | 981 | Georgian lari. |
| GHS | 936 | Ghanaian cedi. |
| GIP | 929 | Gibraltar pound. |
| GMD | 270 | Gambian dalasi. |
| GNF | 324 | Guinean franc. |
| GTQ | 320 | Guatemalan quetzal. |
| GYD | 328 | Guyanese dollar. |
| HKD | 344 | Hong Kong dollar. |
| HNL | 340 | Honduran lempira. |
| HRK | 191 | Croatian kuna. |
| HTG | 332 | Haitian gourde. |
| HUF | 348 | Hungarian forint. |
| IDR | 360 | Indonesian rupiah. |
| ILS | 376 | Israeli new sheqel. |
| INR | 356 | Indian rupee. |
| IQD | 368 | Iraqi dinar. |
| IRR | 364 | Iranian rial. |
| ISK | 352 | Icelandic króna. |
| JMD | 388 | Jamaican dollar. |
| JOD | 400 | Jordanian dinar. |
| JPY | 392 | Japanese yen. |
| KES | 404 | Kenyan shilling. |
| KGS | 417 | Kyrgyzstani som. |
| KHR | 116 | Cambodian riel. |
| KMF | 174 | Comoro franc. |
| KPW | 408 | North Korean won. |
| KRW | 410 | South Korean won. |
| KWD | 414 | Kuwaiti dinar. |
| KYD | 136 | Cayman Islands dollar. |
| KZT | 398 | Kazakhstani tenge. |
| LAK | 418 | Lao kip. |
| LBP | 422 | Lebanese pound. |
| LKR | 144 | Sri Lanka rupee. |
| LRD | 430 | Liberian dollar. |
| LSL | 426 | Lesotho loti. |
| LTL | 440 | Lithuanian litas. |
| LVL | 428 | Latvian lats. |
| LYD | 434 | Libyan dinar. |
| MAD | 504 | Moroccan dirham. |
| MDL | 498 | Moldovan leu. |
| MGA | 969 | Malagasy ariary. |
| MKD | 807 | Macedonian denar. |
| MMK | 104 | Myanma kyat. |
| MNT | 496 | Mongolian tugrik. |
| MOP | 446 | Macanese pataca. |
| MRO | 478 | Mauritanian ouguiya. |
| MUR | 480 | Mauritian rupee. |
| MVR | 462 | Maldivian rufiyaa. |
| MWK | 454 | Malawian kwacha. |
| MXN | 484 | Mexican peso. |
| MYR | 458 | Malaysian ringgit. |
| MZN | 943 | Mozambican metical. |
| NAD | 516 | Namibian dollar. |
| NGN | 566 | Nigerian naira. |
| NIO | 558 | Cordoba oro. |
| NOK | 578 | Norwegian krone. |
| NPR | 524 | Nepalese rupee. |
| NZD | 554 | New Zealand dollar. |
| OMR | 512 | Omani rial. |
| PAB | 590 | Panamanian balboa. |
| PEN | 604 | Peruvian nuevo sol. |
| PGK | 598 | Papua New Guinean kina. |
| PHP | 608 | Philippine peso. |
| PKR | 586 | Pakistani rupee. |
| PLN | 985 | Polish zloty. |
| PYG | 600 | Paraguayan guaraní. |
| QAR | 634 | Qatari rial. |
| RON | 946 | Romanian new leu. |
| RSD | 941 | Serbian dinar. |
| RUB | 643 | Russian rouble. |
| RWF | 646 | Rwandan franc. |
| SAR | 682 | Saudi riyal. |
| SBD | 090 | Solomon Islands dollar. |
| SCR | 690 | Seychelles rupee. |
| SDG | 938 | Sudanese pound. |
| SEK | 752 | Swedish krona/kronor. |
| SGD | 702 | Singapore dollar. |
| SHP | 654 | Saint Helena pound. |
| SLL | 694 | Sierra Leonean leone. |
| SOS | 706 | Somali shilling. |
| SRD | 968 | Surinamese dollar. |
| STD | 678 | São Tomé and Príncipe dobra. |
| SYP | 760 | Syrian pound. |
| SZL | 748 | Lilangeni. |
| THB | 764 | Thai baht. |
| TJS | 972 | Tajikistani somoni. |
| TMT | 934 | Turkmenistani manat. |
| TND | 788 | Tunisian dinar. |
| TOP | 776 | Tongan pa'anga. |
| TRY | 949 | Turkish lira. |
| TTD | 780 | Trinidad and Tobago dollar. |
| TWD | 901 | New Taiwan dollar. |
| TZS | 834 | Tanzanian shilling. |
| UAH | 980 | Ukrainian hryvnia. |
| UGX | 800 | Ugandan shilling. |
| USD | 840 | United States dollar. |
| UYU | 858 | Uruguayan peso. |
| UZS | 860 | Uzbekistan som. |
| VEF | 937 | Venezuelan bolívar fuerte. |
| VND | 704 | Vietnamese Dong. |
| VUV | 548 | Vanuatu vatu. |
| WST | 882 | Samoan tala. |
| XAF | 950 | CFA franc BEAC. |
| XCD | 951 | East Caribbean dollar. |
| XOF | 952 | CFA Franc BCEAO. |
| XPF | 953 | CFP franc. |
| YER | 886 | Yemeni rial. |
| ZAR | 710 | South African rand. |
| ZMK | 894 | Zambian kwacha. |
| ZWL | 932 | Zimbabwe dollar. |

### CurveStyle enumeration

Style or shape of curve.

Table 253 shows all literals of CurveStyle.

Table 253 – Literals of LTDSEquipmentProfile::CurveStyle

| literal | value | description |
| --- | --- | --- |
| 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. |

### FuelType enumeration

Type of fuel.

Table 254 shows all literals of FuelType.

Table 254 – Literals of LTDSEquipmentProfile::FuelType

| literal | value | description |
| --- | --- | --- |
| 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. |
| brownCoalLignite |  | Brown coal lignite. |
| coalDerivedGas |  | Coal derived gas. |
| peat |  | Peat. |
| other |  | Any fuel type not included in the rest of the enumerated value. |

### GeneratorControlSource enumeration

The source of controls for a generating unit.

Table 255 shows all literals of GeneratorControlSource.

Table 255 – Literals of LTDSEquipmentProfile::GeneratorControlSource

| literal | value | description |
| --- | --- | --- |
| unavailable |  | Not available. |
| offAGC |  | Off of automatic generation control (AGC). |
| onAGC |  | On automatic generation control (AGC). |
| plantControl |  | Plant is controlling. |

### HydroEnergyConversionKind enumeration

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

Table 256 shows all literals of HydroEnergyConversionKind.

Table 256 – Literals of LTDSEquipmentProfile::HydroEnergyConversionKind

| literal | value | description |
| --- | --- | --- |
| 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. |

### HydroPlantStorageKind enumeration

The type of hydro power plant.

Table 257 shows all literals of HydroPlantStorageKind.

Table 257 – Literals of LTDSEquipmentProfile::HydroPlantStorageKind

| literal | value | description |
| --- | --- | --- |
| runOfRiver |  | Run of river. |
| pumpedStorage |  | Pumped storage. |
| storage |  | Storage. |

### HydroTurbineKind enumeration

Type of turbine.

Table 258 shows all literals of HydroTurbineKind.

Table 258 – Literals of LTDSEquipmentProfile::HydroTurbineKind

| literal | value | description |
| --- | --- | --- |
| francis |  | Francis. |
| pelton |  | Pelton. |
| kaplan |  | Kaplan. |

### (European) LimitKind enumeration

Limit kinds.

Table 259 shows all literals of LimitKind.

Table 259 – Literals of LTDSEquipmentProfile::LimitKind

| literal | value | description |
| --- | --- | --- |
| patl |  | (European) The Permanent Admissible Transmission Loading (PATL) is the loading in amperes, MVA or MW that can be accepted by a network branch for an unlimited duration without any risk for the material.  The OperationnalLimitType.isInfiniteDuration is set to true. There shall be only one OperationalLimitType of kind PATL per OperationalLimitSet if the PATL is ApparentPowerLimit, ActivePowerLimit, or CurrentLimit for a given Terminal or Equipment. |
| patlt |  | (European) Permanent Admissible Transmission Loading Threshold (PATLT) is a value in engineering units defined for PATL and calculated using a 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 |  | (European) Temporarily Admissible Transmission Loading (TATL) which is the loading in amperes, 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 to define several TATL limit types. Hence multiple TATL limit values may exist having different durations. |
| tc |  | (European) 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 if the OperationalLimitType.acceptableDuration is exchanged. Only one limit value exists for the TC type. |
| tct |  | (European) Tripping Current Threshold (TCT) is a value in engineering units defined for TC and calculated using a 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 |  | (European) 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 |  | (European) 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. |
| operationalVoltageLimit |  | (European) Operational voltage limit. |
| alarmVoltage |  | (European) Voltage alarm. |
| warningVoltage |  | (European) Voltage warning. |
| stability |  | (European) Stability. |

### OperationalLimitDirectionKind enumeration

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

Table 260 shows all literals of OperationalLimitDirectionKind.

Table 260 – Literals of LTDSEquipmentProfile::OperationalLimitDirectionKind

| literal | value | description |
| --- | --- | --- |
| 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. |

### PhaseCode enumeration

An unordered enumeration of phase identifiers. Allows designation of phases for both transmission and distribution equipment, circuits and loads. The enumeration, by itself, does not describe how the phases are connected together or connected to ground. Ground is not explicitly denoted as a phase.

Residential and small commercial loads are often served from single-phase, or split-phase, secondary circuits. For 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.

The integer values are from IEC 61968-9 to support revenue metering applications.

Table 261 shows all literals of PhaseCode.

Table 261 – Literals of LTDSEquipmentProfile::PhaseCode

| literal | value | description |
| --- | --- | --- |
| ABCN | 225 | Phases A, B, C, and N. |
| ABC | 224 | Phases A, B, and C. |
| ABN | 193 | Phases A, B, and neutral. |
| ACN | 41 | Phases A, C and neutral. |
| BCN | 97 | Phases B, C, and neutral. |
| AB | 132 | Phases A and B. |
| AC | 96 | Phases A and C. |
| BC | 66 | Phases B and C. |
| AN | 129 | Phases A and neutral. |
| BN | 65 | Phases B and neutral. |
| CN | 33 | Phases C and neutral. |
| A | 128 | Phase A. |
| B | 64 | Phase B. |
| C | 32 | Phase C. |
| N | 16 | Neutral phase. |
| s1N | 528 | Secondary phase 1 and neutral. |
| s2N | 272 | Secondary phase 2 and neutral. |
| s12N | 784 | Secondary phases 1, 2, and neutral. |
| s1 | 512 | Secondary phase 1. |
| s2 | 256 | Secondary phase 2. |
| s12 | 768 | Secondary phase 1 and 2. |
| none | 0 | No phases specified. |
| X | 1024 | Unknown non-neutral phase. |
| XY | 3072 | Two unknown non-neutral phases. |
| XN | 1040 | Unknown non-neutral phase plus neutral. |
| XYN | 3088 | Two unknown non-neutral phases plus neutral. |

### RegulatingControlModeKind enumeration

The kind of regulation model. For example regulating voltage, reactive power, active power, etc.

Table 262 shows all literals of RegulatingControlModeKind.

Table 262 – Literals of LTDSEquipmentProfile::RegulatingControlModeKind

| literal | value | description |
| --- | --- | --- |
| 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. |

### SynchronousMachineKind enumeration

Synchronous machine type.

Table 263 shows all literals of SynchronousMachineKind.

Table 263 – Literals of LTDSEquipmentProfile::SynchronousMachineKind

| literal | value | description |
| --- | --- | --- |
| generator |  | Indicates the synchronous machine can operate as a generator. |
| condenser |  | Indicates the synchronous machine can operate as a condenser. |
| generatorOrCondenser |  | Indicates the synchronous machine can operate as a generator or as a condenser. |
| motor |  | Indicates the synchronous machine can operate as a motor. |
| generatorOrMotor |  | Indicates the synchronous machine can operate as a generator or as a motor. |
| motorOrCondenser |  | Indicates the synchronous machine can operate as a motor or as a condenser. |
| generatorOrCondenserOrMotor |  | Indicates the synchronous machine can operate as a generator or as a condenser or as a motor. |

### UnitMultiplier enumeration

The unit multipliers defined for the CIM. When applied to unit symbols, the unit symbol is treated as a derived unit. Regardless of the contents of the unit symbol text, the unit symbol shall be treated as if it were a single-character unit symbol. Unit symbols should not contain multipliers, and it should be left to the multiplier to define the multiple for an entire data type.

For example, if a unit symbol is "m2Pers" and the multiplier is "k", then the value is k(m\*\*2/s), and the multiplier applies to the entire final value, not to any individual part of the value. This can be conceptualized by substituting a derived unit symbol for the unit type. If one imagines that the symbol "Þ" represents the derived unit "m2Pers", then applying the multiplier "k" can be conceptualized simply as "kÞ".

For example, the SI unit for mass is "kg" and not "g". If the unit symbol is defined as "kg", then the multiplier is applied to "kg" as a whole and does not replace the "k" in front of the "g". In this case, the multiplier of "m" would be used with the unit symbol of "kg" to represent one gram. As a text string, this violates the instructions in IEC 80000-1. However, because the unit symbol in CIM is treated as a derived unit instead of as an SI unit, it makes more sense to conceptualize the "kg" as if it were replaced by one of the proposed replacements for the SI mass symbol. If one imagines that the "kg" were replaced by a symbol "Þ", then it is easier to conceptualize the multiplier "m" as creating the proper unit "mÞ", and not the forbidden unit "mkg".

Table 264 shows all literals of UnitMultiplier.

Table 264 – Literals of LTDSEquipmentProfile::UnitMultiplier

| literal | value | description |
| --- | --- | --- |
| y | -24 | Yocto 10\*\*-24. |
| z | -21 | Zepto 10\*\*-21. |
| a | -18 | Atto 10\*\*-18. |
| f | -15 | Femto 10\*\*-15. |
| p | -12 | Pico 10\*\*-12. |
| n | -9 | Nano 10\*\*-9. |
| micro | -6 | Micro 10\*\*-6. |
| m | -3 | Milli 10\*\*-3. |
| c | -2 | Centi 10\*\*-2. |
| d | -1 | Deci 10\*\*-1. |
| none | 0 | No multiplier or equivalently multiply by 1. |
| da | 1 | Deca 10\*\*1. |
| h | 2 | Hecto 10\*\*2. |
| k | 3 | Kilo 10\*\*3. |
| M | 6 | Mega 10\*\*6. |
| G | 9 | Giga 10\*\*9. |
| T | 12 | Tera 10\*\*12. |
| P | 15 | Peta 10\*\*15. |
| E | 18 | Exa 10\*\*18. |
| Z | 21 | Zetta 10\*\*21. |
| Y | 24 | Yotta 10\*\*24. |

### UnitSymbol enumeration

The derived units defined for usage in the CIM. In some cases, the derived unit is equal to an SI unit. Whenever possible, the standard derived symbol is used instead of the formula for the derived unit. For example, the unit symbol Farad is defined as "F" instead of "CPerV". In cases where a standard symbol does not exist for a derived unit, the formula for the unit is used as the unit symbol. For example, density does not have a standard symbol and so it is represented as "kgPerm3". With the exception of the "kg", which is an SI unit, the unit symbols do not contain multipliers and therefore represent the base derived unit to which a multiplier can be applied as a whole.

Every unit symbol is treated as an unparseable text as if it were a single-letter symbol. The meaning of each unit symbol is defined by the accompanying descriptive text and not by the text contents of the unit symbol.

To allow the widest possible range of serializations without requiring special character handling, several substitutions are made which deviate from the format described in IEC 80000-1. The division symbol "/" is replaced by the letters "Per". Exponents are written in plain text after the unit as "m3" instead of being formatted as "m" with a superscript of 3 or introducing a symbol as in "m^3". The degree symbol "°" is replaced with the letters "deg". Any clarification of the meaning for a substitution is included in the description for the unit symbol.

Non-SI units are included in list of unit symbols to allow sources of data to be correctly labelled with their non-SI units (for example, a GPS sensor that is reporting numbers that represent feet instead of meters). This allows software to use the unit symbol information correctly convert and scale the raw data of those sources into SI-based units.

The integer values are used for harmonization with IEC 61850.

Table 265 shows all literals of UnitSymbol.

Table 265 – Literals of LTDSEquipmentProfile::UnitSymbol

| literal | value | description |
| --- | --- | --- |
| none | 0 | Dimension less quantity, e.g. count, per unit, etc. |
| m | 2 | Length in metres. |
| kg | 3 | Mass in kilograms. Note: multiplier “k” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| s | 4 | Time in seconds. |
| A | 5 | Current in amperes. |
| K | 6 | Temperature in kelvins. |
| mol | 7 | Amount of substance in moles. |
| cd | 8 | Luminous intensity in candelas. |
| deg | 9 | Plane angle in degrees. |
| rad | 10 | Plane angle in radians (m/m). |
| sr | 11 | Solid angle in steradians (m2/m2). |
| Gy | 21 | Absorbed dose in grays (J/kg). |
| Bq | 22 | Radioactivity in becquerels (1/s). |
| degC | 23 | 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 from coulomb the symbol used in the UML is degC. The reason for not using °C is that the special character ° is difficult to manage in software. |
| Sv | 24 | Dose equivalent in sieverts (J/kg). |
| F | 25 | Electric capacitance in farads (C/V). |
| C | 26 | Electric charge in coulombs (A·s). |
| S | 27 | Conductance in siemens. |
| H | 28 | Electric inductance in henrys (Wb/A). |
| V | 29 | Electric potential in volts (W/A). |
| ohm | 30 | Electric resistance in ohms (V/A). |
| J | 31 | Energy in joules (N·m = C·V = W·s). |
| N | 32 | Force in newtons (kg·m/s²). |
| Hz | 33 | Frequency in hertz (1/s). |
| lx | 34 | Illuminance in lux (lm/m²). |
| lm | 35 | Luminous flux in lumens (cd·sr). |
| Wb | 36 | Magnetic flux in webers (V·s). |
| T | 37 | Magnetic flux density in teslas (Wb/m2). |
| W | 38 | Real power in watts (J/s). Electrical power may have real and reactive components. The real portion of electrical power (I²R or VIcos(phi)), is expressed in Watts. See also apparent power and reactive power. |
| Pa | 39 | Pressure in pascals (N/m²). Note: the absolute or relative measurement of pressure is implied with this entry. See below for more explicit forms. |
| m2 | 41 | Area in square metres (m²). |
| m3 | 42 | Volume in cubic metres (m³). |
| mPers | 43 | Velocity in metres per second (m/s). |
| mPers2 | 44 | Acceleration in metres per second squared (m/s²). |
| m3Pers | 45 | Volumetric flow rate in cubic metres per second (m³/s). |
| mPerm3 | 46 | Fuel efficiency in metres per cubic metres (m/m³). |
| kgm | 47 | Moment of mass in kilogram metres (kg·m) (first moment of mass). Note: multiplier “k” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| kgPerm3 | 48 | Density in kilogram/cubic metres (kg/m³). Note: multiplier “k” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| m2Pers | 49 | Viscosity in square metres / second (m²/s). |
| WPermK | 50 | Thermal conductivity in watt/metres kelvin. |
| JPerK | 51 | Heat capacity in joules/kelvin. |
| ppm | 52 | Concentration in parts per million. |
| rotPers | 53 | Rotations per second (1/s). See also Hz (1/s). |
| radPers | 54 | Angular velocity in radians per second (rad/s). |
| WPerm2 | 55 | Heat flux density, irradiance, watts per square metre. |
| JPerm2 | 56 | Insulation energy density, joules per square metre or watt second per square metre. |
| SPerm | 57 | Conductance per length (F/m). |
| KPers | 58 | Temperature change rate in kelvins per second. |
| PaPers | 59 | Pressure change rate in pascals per second. |
| JPerkgK | 60 | Specific heat capacity, specific entropy, joules per kilogram Kelvin. |
| VA | 61 | Apparent power in volt amperes. See also real power and reactive power. |
| VAr | 63 | Reactive power in volt amperes reactive. The “reactive” or “imaginary” component of electrical power (VIsin(phi)). (See also real power and apparent power).  Note: Different meter designs use different methods to arrive at their results. Some meters may compute reactive power as an arithmetic value, while others compute the value vectorially. The data consumer should determine the method in use and the suitability of the measurement for the intended purpose. |
| cosPhi | 65 | Power factor, dimensionless.  Note 1: This definition of power factor only holds for balanced systems. See the alternative definition under code 153.  Note 2 : Beware of differing sign conventions in use between the IEC and EEI. It is assumed that the data consumer understands the type of meter in use and the sign convention in use by the utility. |
| Vs | 66 | Volt seconds (Ws/A). |
| V2 | 67 | Volt squared (W²/A²). |
| As | 68 | Ampere seconds (A·s). |
| A2 | 69 | Amperes squared (A²). |
| A2s | 70 | Ampere squared time in square amperes (A²s). |
| VAh | 71 | Apparent energy in volt ampere hours. |
| Wh | 72 | Real energy in watt hours. |
| VArh | 73 | Reactive energy in volt ampere reactive hours. |
| VPerHz | 74 | Magnetic flux in volt per hertz. |
| HzPers | 75 | Rate of change of frequency in hertz per second. |
| character | 76 | Number of characters. |
| charPers | 77 | Data rate (baud) in characters per second. |
| kgm2 | 78 | Moment of mass in kilogram square metres (kg·m²) (Second moment of mass, commonly called the moment of inertia). Note: multiplier “k” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| dB | 79 | Sound pressure level in decibels. Note: multiplier “d” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| WPers | 81 | Ramp rate in watts per second. |
| lPers | 82 | Volumetric flow rate in litres per second. |
| dBm | 83 | Power level (logarithmic ratio of signal strength , Bel-mW), normalized to 1mW. Note: multiplier “d” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| h | 84 | Time in hours, hour = 60 min = 3600 s. |
| min | 85 | Time in minutes, minute = 60 s. |
| Q | 100 | Quantity power, Q. |
| Qh | 101 | Quantity energy, Qh. |
| ohmm | 102 | Resistivity, ohm metres, (rho). |
| APerm | 103 | A/m, magnetic field strength, amperes per metre. |
| V2h | 104 | Volt-squared hour, volt-squared-hours. |
| A2h | 105 | Ampere-squared hour, ampere-squared hour. |
| Ah | 106 | Ampere-hours, ampere-hours. |
| count | 111 | Amount of substance, Counter value. |
| ft3 | 119 | Volume, cubic feet. |
| m3Perh | 125 | Volumetric flow rate, cubic metres per hour. |
| gal | 128 | Volume in gallons, US gallon (1 gal = 231 in3 = 128 fl ounce). |
| Btu | 132 | Energy, British Thermal Units. |
| l | 134 | Volume in litres, litre = dm3 = m3/1000. |
| lPerh | 137 | Volumetric flow rate, litres per hour. |
| lPerl | 143 | Concentration, The ratio of the volume of a solute divided by the volume of the solution. Note: Users may need use a prefix such a ‘µ’ to express a quantity such as ‘µL/L’. |
| gPerg | 144 | Concentration, The ratio of the mass of a solute divided by the mass of the solution. Note: Users may need use a prefix such a ‘µ’ to express a quantity such as ‘µg/g’. |
| molPerm3 | 145 | Concentration, The amount of substance concentration, (c), the amount of solvent in moles divided by the volume of solution in m³. |
| molPermol | 146 | Concentration, Molar fraction, the ratio of the molar amount of a solute divided by the molar amount of the solution. |
| molPerkg | 147 | Concentration, Molality, the amount of solute in moles and the amount of solvent in kilograms. |
| sPers | 149 | Time, Ratio of time. Note: Users may need to supply a prefix such as ‘µ’ to show rates such as ‘µs/s’. |
| HzPerHz | 150 | Frequency, rate of frequency change. Note: Users may need to supply a prefix such as ‘m’ to show rates such as ‘mHz/Hz’. |
| VPerV | 151 | Voltage, ratio of voltages. Note: Users may need to supply a prefix such as ‘m’ to show rates such as ‘mV/V’. |
| APerA | 152 | Current, ratio of amperages. Note: Users may need to supply a prefix such as ‘m’ to show rates such as ‘mA/A’. |
| VPerVA | 153 | Power factor, PF, the ratio of the active power to the apparent power. Note: The sign convention used for power factor will differ between IEC meters and EEI (ANSI) meters. It is assumed that the data consumers understand the type of meter being used and agree on the sign convention in use at any given utility. |
| rev | 154 | Amount of rotation, revolutions. |
| kat | 158 | Catalytic activity, katal = mol / s. |
| JPerkg | 165 | Specific energy, Joules / kg. |
| m3Uncompensated | 166 | Volume, cubic metres, with the value uncompensated for weather effects. |
| m3Compensated | 167 | Volume, cubic metres, with the value compensated for weather effects. |
| WPerW | 168 | Signal Strength, ratio of power. Note: Users may need to supply a prefix such as ‘m’ to show rates such as ‘mW/W’. |
| therm | 169 | Energy, therms. |
| onePerm | 173 | Wavenumber, reciprocal metres, (1/m). |
| m3Perkg | 174 | Specific volume, cubic metres per kilogram, v. |
| Pas | 175 | Dynamic viscosity, pascal seconds. |
| Nm | 176 | Moment of force, newton metres. |
| NPerm | 177 | Surface tension, newton per metre. |
| radPers2 | 178 | Angular acceleration, radians per second squared. |
| JPerm3 | 181 | Energy density, joules per cubic metre. |
| VPerm | 182 | Electric field strength, volts per metre. |
| CPerm3 | 183 | Electric charge density, coulombs per cubic metre. |
| CPerm2 | 184 | Surface charge density, coulombs per square metre. |
| FPerm | 185 | Permittivity, farads per metre. |
| HPerm | 186 | Permeability, henrys per metre. |
| JPermol | 187 | Molar energy, joules per mole. |
| JPermolK | 188 | Molar entropy, molar heat capacity, joules per mole kelvin. |
| CPerkg | 189 | Exposure (x rays), coulombs per kilogram. |
| GyPers | 190 | Absorbed dose rate, grays per second. |
| WPersr | 191 | Radiant intensity, watts per steradian. |
| WPerm2sr | 192 | Radiance, watts per square metre steradian. |
| katPerm3 | 193 | Catalytic activity concentration, katals per cubic metre. |
| d | 195 | Time in days, day = 24 h = 86400 s. |
| anglemin | 196 | Plane angle, minutes. |
| anglesec | 197 | Plane angle, seconds. |
| ha | 198 | Area, hectares. |
| tonne | 199 | Mass in tons, “tonne” or “metric ton” (1000 kg = 1 Mg). |
| bar | 214 | Pressure in bars, (1 bar = 100 kPa). |
| mmHg | 215 | Pressure, millimetres of mercury (1 mmHg is approximately 133.3 Pa). |
| M | 217 | Length, nautical miles (1 M = 1852 m). |
| kn | 219 | Speed, knots (1 kn = 1852/3600) m/s. |
| Mx | 276 | Magnetic flux, maxwells (1 Mx = 10-8 Wb). |
| G | 277 | Magnetic flux density, gausses (1 G = 10-4 T). |
| Oe | 278 | Magnetic field in oersteds, (1 Oe = (103/4p) A/m). |
| Vh | 280 | Volt-hour, Volt hours. |
| WPerA |  | Active power per current flow, watts per Ampere. |
| onePerHz |  | Reciprocal of frequency (1/Hz). |
| VPerVAr |  | Power factor, PF, the ratio of the active power to the apparent power. Note: The sign convention used for power factor will differ between IEC meters and EEI (ANSI) meters. It is assumed that the data consumers understand the type of meter being used and agree on the sign convention in use at any given utility. |
| ohmPerm | 86 | Electric resistance per length in ohms per metre ((V/A)/m). |
| kgPerJ |  | Weight per energy in kilograms per joule (kg/J). Note: multiplier “k” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| JPers |  | Energy rate in joules per second (J/s). |

### WindGenUnitKind enumeration

Kind of wind generating unit.

Table 266 shows all literals of WindGenUnitKind.

Table 266 – Literals of LTDSEquipmentProfile::WindGenUnitKind

| literal | value | description |
| --- | --- | --- |
| offshore |  | The wind generating unit is located offshore. |
| onshore |  | The wind generating unit is located onshore. |

### WindingConnection enumeration

Winding connection type.

Table 267 shows all literals of WindingConnection.

Table 267 – Literals of LTDSEquipmentProfile::WindingConnection

| literal | value | description |
| --- | --- | --- |
| 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. |

### ActivePower datatype

Product of RMS value of the voltage and the RMS value of the in-phase component of the current.

Table 268 shows all attributes of ActivePower.

Table 268 – Attributes of LTDSEquipmentProfile::ActivePower

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML64) |  |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=M) |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=W) |

### ActivePowerPerFrequency datatype

Active power variation with frequency.

Table 269 shows all attributes of ActivePowerPerFrequency.

Table 269 – Attributes of LTDSEquipmentProfile::ActivePowerPerFrequency

| name | mult | type | description |
| --- | --- | --- | --- |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=M) |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=WPers) |
| value | 0..1 | [Float](#UML64) |  |

### AngleDegrees datatype

Measurement of angle in degrees.

Table 270 shows all attributes of AngleDegrees.

Table 270 – Attributes of LTDSEquipmentProfile::AngleDegrees

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML64) |  |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=deg) |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=none) |

### ReactancePerLength datatype

Reactance (imaginary part of impedance) per unit of length, at rated frequency.

Table 271 shows all attributes of ReactancePerLength.

Table 271 – Attributes of LTDSEquipmentProfile::ReactancePerLength

| name | mult | type | description |
| --- | --- | --- | --- |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=none) |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=ohmPerm) |
| value | 0..1 | [Float](#UML64) |  |

### ConductancePerLength datatype

Real part of admittance per unit of length.

Table 272 shows all attributes of ConductancePerLength.

Table 272 – Attributes of LTDSEquipmentProfile::ConductancePerLength

| name | mult | type | description |
| --- | --- | --- | --- |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=none) |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=SPerm) |
| value | 0..1 | [Float](#UML64) |  |

### ResistancePerLength datatype

Resistance (real part of impedance) per unit of length.

Table 273 shows all attributes of ResistancePerLength.

Table 273 – Attributes of LTDSEquipmentProfile::ResistancePerLength

| name | mult | type | description |
| --- | --- | --- | --- |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=none) |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=ohmPerm) |
| value | 0..1 | [Float](#UML64) |  |

### SusceptancePerLength datatype

Imaginary part of admittance per unit of length.

Table 274 shows all attributes of SusceptancePerLength.

Table 274 – Attributes of LTDSEquipmentProfile::SusceptancePerLength

| name | mult | type | description |
| --- | --- | --- | --- |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=none) |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=SPerm) |
| value | 0..1 | [Float](#UML64) |  |

### ApparentPower datatype

Product of the RMS value of the voltage and the RMS value of the current.

Table 275 shows all attributes of ApparentPower.

Table 275 – Attributes of LTDSEquipmentProfile::ApparentPower

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML64) |  |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=M) |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=VA) |

### Conductance datatype

Factor by which voltage must be multiplied to give corresponding power lost from a circuit. Real part of admittance.

Table 276 shows all attributes of Conductance.

Table 276 – Attributes of LTDSEquipmentProfile::Conductance

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML64) |  |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=S) |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=none) |

### CurrentFlow datatype

Electrical current with sign convention: positive flow is out of the conducting equipment into the connectivity node. Can be both AC and DC.

Table 277 shows all attributes of CurrentFlow.

Table 277 – Attributes of LTDSEquipmentProfile::CurrentFlow

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML64) |  |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=none) |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=A) |

### Frequency datatype

Cycles per second.

Table 278 shows all attributes of Frequency.

Table 278 – Attributes of LTDSEquipmentProfile::Frequency

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML64) |  |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=Hz) |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=none) |

### Length datatype

Unit of length. It shall be a positive value or zero.

Table 279 shows all attributes of Length.

Table 279 – Attributes of LTDSEquipmentProfile::Length

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML64) |  |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=m) |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=k) |

### Money datatype

Amount of money.

Table 280 shows all attributes of Money.

Table 280 – Attributes of LTDSEquipmentProfile::Money

| name | mult | type | description |
| --- | --- | --- | --- |
| unit | 0..1 | [Currency](#UML17) |  |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=none) |
| value | 0..1 | [Decimal](#UML62) |  |

### PerCent datatype

Percentage on a defined base. For example, specify as 100 to indicate at the defined base.

Table 281 shows all attributes of PerCent.

Table 281 – Attributes of LTDSEquipmentProfile::PerCent

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML64) | Normally 0 to 100 on a defined base. |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=none) |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=none) |

### Reactance datatype

Reactance (imaginary part of impedance), at rated frequency.

Table 282 shows all attributes of Reactance.

Table 282 – Attributes of LTDSEquipmentProfile::Reactance

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML64) |  |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=ohm) |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=none) |

### ReactivePower datatype

Product of RMS value of the voltage and the RMS value of the quadrature component of the current.

Table 283 shows all attributes of ReactivePower.

Table 283 – Attributes of LTDSEquipmentProfile::ReactivePower

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML64) |  |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=VAr) |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=M) |

### RealEnergy datatype

Real electrical energy.

Table 284 shows all attributes of RealEnergy.

Table 284 – Attributes of LTDSEquipmentProfile::RealEnergy

| name | mult | type | description |
| --- | --- | --- | --- |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=M) |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=Wh) |
| value | 0..1 | [Float](#UML64) |  |

### Resistance datatype

Resistance (real part of impedance).

Table 285 shows all attributes of Resistance.

Table 285 – Attributes of LTDSEquipmentProfile::Resistance

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML64) |  |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=ohm) |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=none) |

### RotationSpeed datatype

Number of revolutions per second.

Table 286 shows all attributes of RotationSpeed.

Table 286 – Attributes of LTDSEquipmentProfile::RotationSpeed

| name | mult | type | description |
| --- | --- | --- | --- |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=none) |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=Hz) |
| value | 0..1 | [Float](#UML64) |  |

### Seconds datatype

Time, in seconds.

Table 287 shows all attributes of Seconds.

Table 287 – Attributes of LTDSEquipmentProfile::Seconds

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML64) | Time, in seconds |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=s) |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=none) |

### Susceptance datatype

Imaginary part of admittance.

Table 288 shows all attributes of Susceptance.

Table 288 – Attributes of LTDSEquipmentProfile::Susceptance

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML64) |  |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=S) |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=none) |

### Voltage datatype

Electrical voltage, can be both AC and DC.

Table 289 shows all attributes of Voltage.

Table 289 – Attributes of LTDSEquipmentProfile::Voltage

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML64) |  |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=k) |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=V) |

### VoltagePerReactivePower datatype

Voltage variation with reactive power.

Table 290 shows all attributes of VoltagePerReactivePower.

Table 290 – Attributes of LTDSEquipmentProfile::VoltagePerReactivePower

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML64) |  |
| unit | 0..1 | [UnitSymbol](#UML30) | (const=VPerVAr) |
| multiplier | 0..1 | [UnitMultiplier](#UML29) | (const=none) |

### (profcim) IRI primitive

An IRI (Internationalized Resource Identifier) within an RDF graph is a Unicode string that conforms to the syntax defined in RFC 3987.

IRIs in the RDF abstract syntax must be absolute, and may contain a fragment identifier.

IRI equality: Two IRIs are equal if and only if they are equivalent under Simple String Comparison according to section 5.1 of [RFC3987]. Further normalization must not be performed when comparing IRIs for equality.

IRIs are a generalization of URIs [RFC3986] that permits a wider range of Unicode characters. Every absolute URI and URL is an IRI, but not every IRI is an URI. When IRIs are used in operations that are only defined for URIs, they must first be converted according to the mapping defined in section 3.1 of [RFC3987]. A notable example is retrieval over the HTTP protocol. The mapping involves UTF-8 encoding of non-ASCII characters, %-encoding of octets not allowed in URIs, and Punycode-encoding of domain names.

### Boolean primitive

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

### (profcim) StringIRI primitive

An IRI (Internationalized Resource Identifier) within an RDF graph is a Unicode string that conforms to the syntax defined in RFC 3987.

The primitive is serialized as literal without language support.

IRIs in the RDF abstract syntax must be absolute, and may contain a fragment identifier.

IRI equality: Two IRIs are equal if and only if they are equivalent under Simple String Comparison according to section 5.1 of [RFC3987]. Further normalization must not be performed when comparing IRIs for equality.

IRIs are a generalization of URIs [RFC3986] that permits a wider range of Unicode characters. Every absolute URI and URL is an IRI, but not every IRI is an URI. When IRIs are used in operations that are only defined for URIs, they must first be converted according to the mapping defined in section 3.1 of [RFC3987]. A notable example is retrieval over the HTTP protocol. The mapping involves UTF-8 encoding of non-ASCII characters, %-encoding of octets not allowed in URIs, and Punycode-encoding of domain names.

### (profcim) StringFixedLanguage primitive

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

The primitive is serialized as literal without language support.

### (profcim) URL primitive

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

### Date primitive

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

### Decimal primitive

Decimal is the base-10 notational system for representing real numbers.

### DateTime primitive

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

### Float primitive

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

### Integer primitive

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

### MonthDay primitive

MonthDay format as "--mm-dd", which conforms with XSD data type gMonthDay.

### String primitive

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

### Package AdditionalClasses

#### General

#### PerLengthImpedance

Inheritance path = [PerLengthLineParameter](#UML2014) : [IdentifiedObject](#UML12)

Common type for per-length impedance electrical catalogues.

Table 291 shows all attributes of PerLengthImpedance.

Table 291 – Attributes of AdditionalClasses::PerLengthImpedance

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

#### PerLengthLineParameter

Inheritance path = [IdentifiedObject](#UML12)

Common type for per-length electrical catalogues describing line parameters.

Table 292 shows all attributes of PerLengthLineParameter.

Table 292 – Attributes of AdditionalClasses::PerLengthLineParameter

| name | mult | type | description |
| --- | --- | --- | --- |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

#### PerLengthSequenceImpedance

Inheritance path = [PerLengthImpedance](#UML2015) : [PerLengthLineParameter](#UML2014) : [IdentifiedObject](#UML12)

Sequence impedance and admittance parameters per unit length, for transposed lines of 1, 2, or 3 phases. For 1-phase lines, define x=x0=xself. For 2-phase lines, define x=xs-xm and x0=xs+xm.

Table 293 shows all attributes of PerLengthSequenceImpedance.

Table 293 – Attributes of AdditionalClasses::PerLengthSequenceImpedance

| name | mult | type | description |
| --- | --- | --- | --- |
| bch | 0..1 | [SusceptancePerLength](#UML39) | Positive sequence shunt (charging) susceptance, per unit of length. |
| gch | 0..1 | [ConductancePerLength](#UML37) | Positive sequence shunt (charging) conductance, per unit of length. |
| r | 0..1 | [ResistancePerLength](#UML38) | Positive sequence series resistance, per unit of length. |
| x | 0..1 | [ReactancePerLength](#UML36) | Positive sequence series reactance, per unit of length. |
| description | 0..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| mRID | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |
| name | 1..1 | [String](#UML67) | inherited from: [IdentifiedObject](#UML12) |

## Requirements and constraints

# LTDS Short circuit profile

## Detailed specification

### General

This is the LTDS short circuit profile.

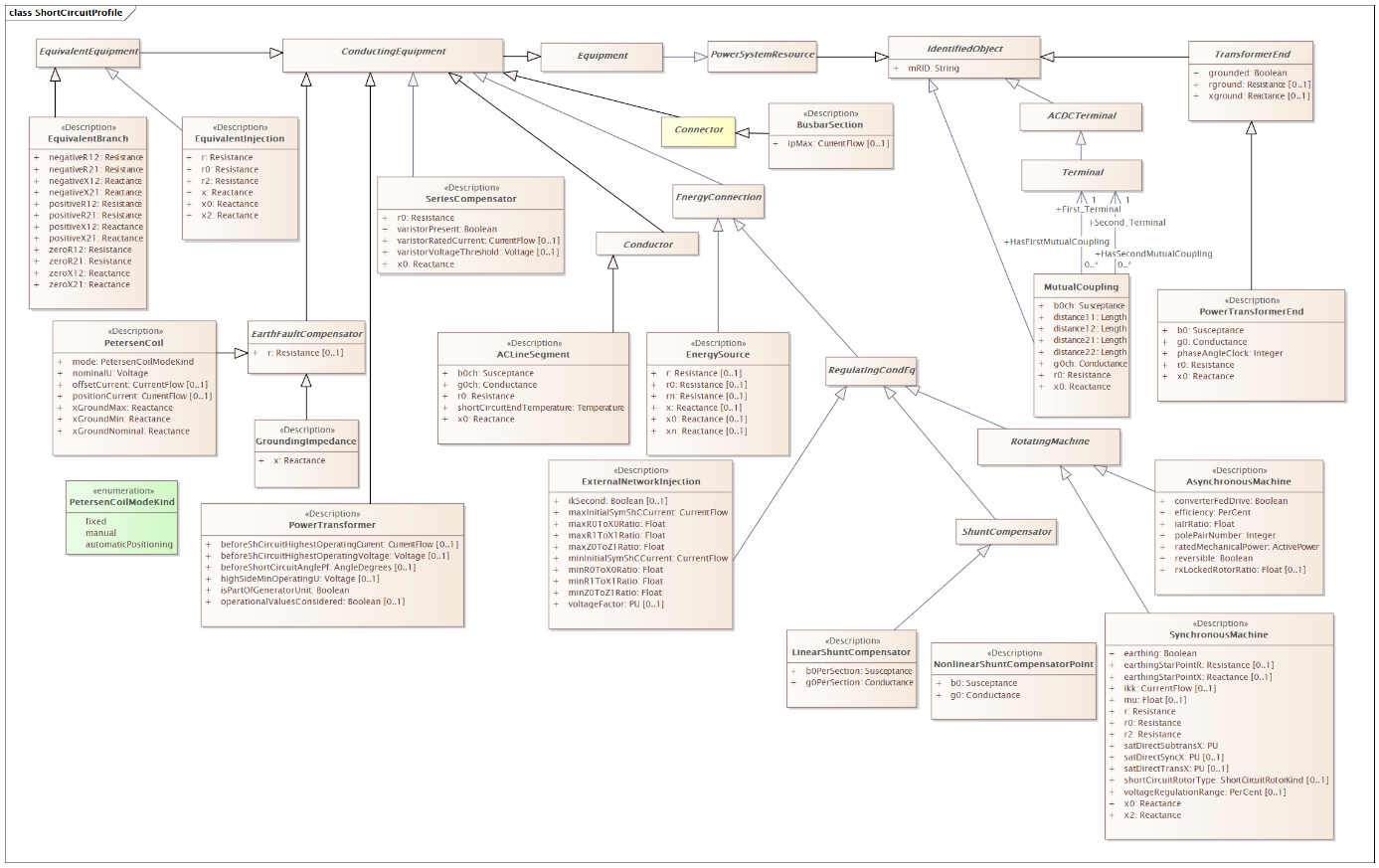


Figure 24 – Class diagram LTDSShortCircuitProfile::ShortCircuitProfile

Figure 24: The diagram shows all classes related to equipment short circuit profile.

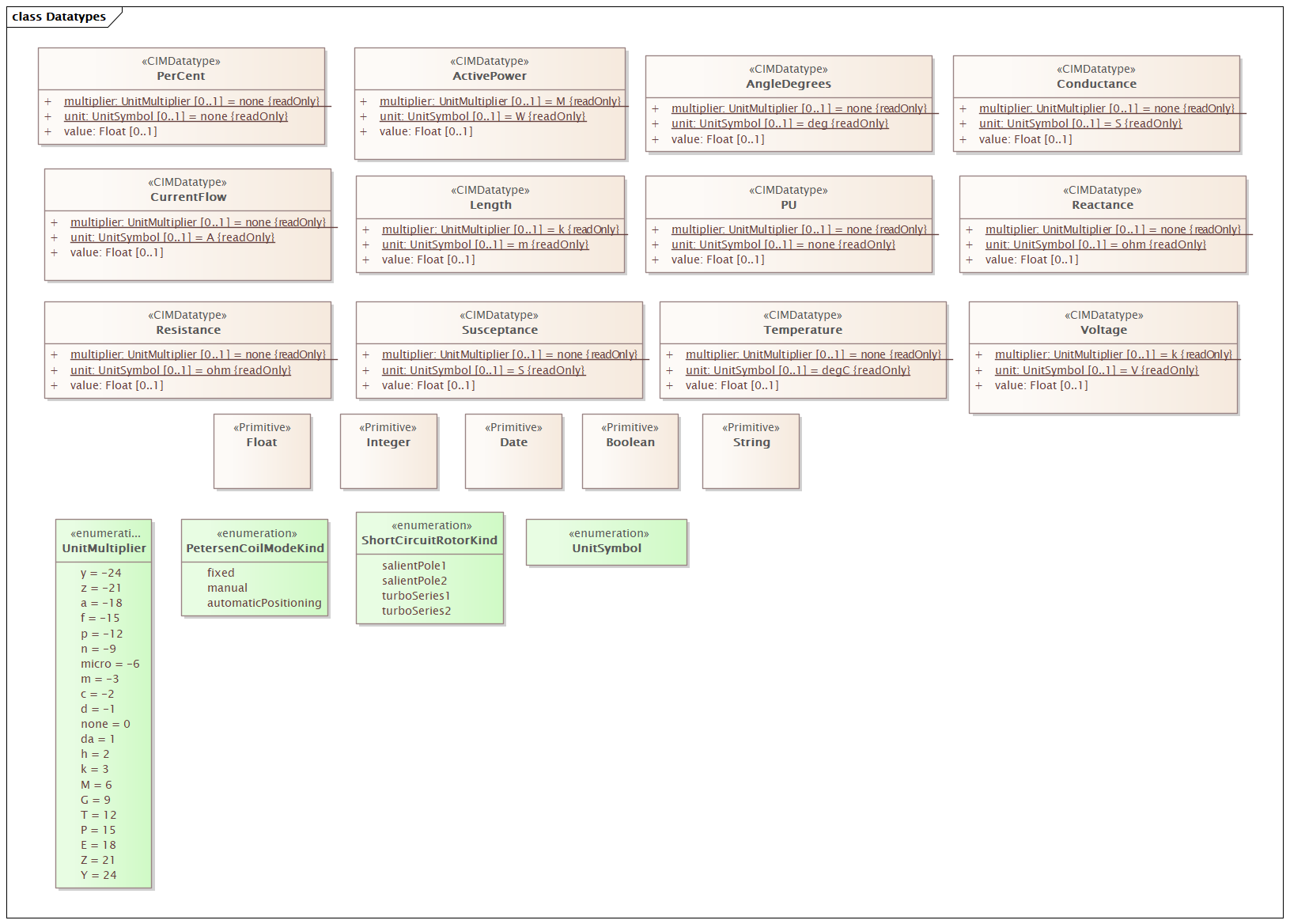


Figure 25 – Class diagram LTDSShortCircuitProfile::Datatypes

Figure 25: The diagram shows datatypes that are used by classes in the profile. Stereotypes are used to describe the datatypes. The following stereotypes are defined:

<<enumeration>> A list of permissible constant values.

<<Primitive>> The most basic data types used to compose all other data types.

<<CIMDatatype>> A datatype that contains a value attribute, an optional unit of measure and a unit multiplier. The unit and multiplier may be specified as a static variable initialized to the allowed value.

<<Compound>> A composite of Primitive, enumeration, CIMDatatype or other Compound classes, as long as the Compound classes do not recurse.

For all datatypes both positive and negative values are allowed unless stated otherwise for a particular datatype.

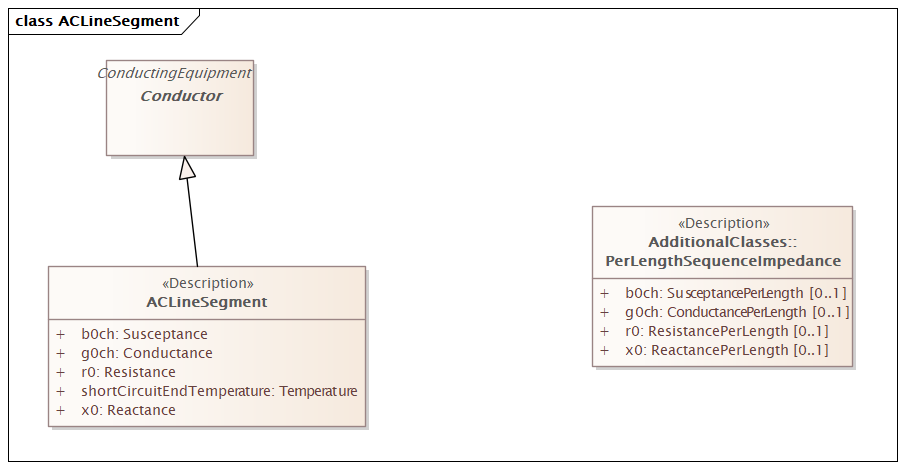


Figure 26 – Class diagram LTDSShortCircuitProfile::ACLineSegment

Figure 26:

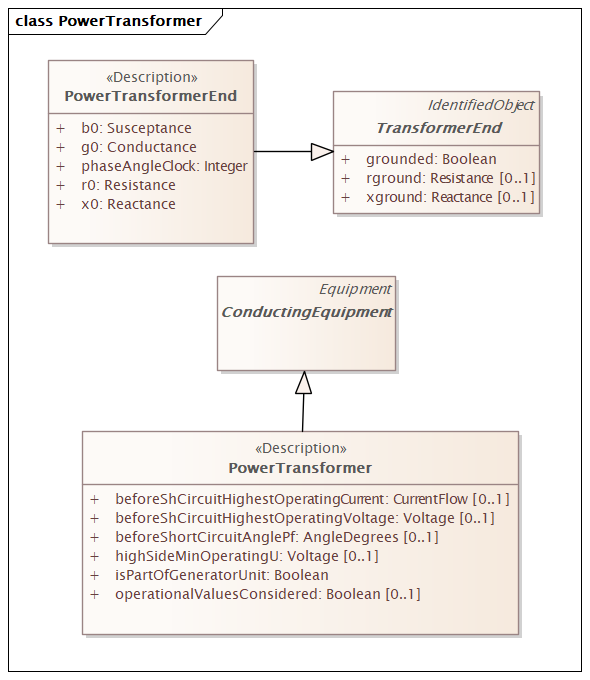


Figure 27 – Class diagram LTDSShortCircuitProfile::PowerTransformer

Figure 27:

### (abstract) ACDCTerminal

Inheritance path = [IdentifiedObject](#UML68)

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

Table 294 shows all attributes of ACDCTerminal.

Table 294 – Attributes of LTDSShortCircuitProfile::ACDCTerminal

| name | mult | type | description |
| --- | --- | --- | --- |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (Description) ACLineSegment

Inheritance path = [Conductor](#UML2025) : [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

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 three phase 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.

Table 295 shows all attributes of ACLineSegment.

Table 295 – Attributes of LTDSShortCircuitProfile::ACLineSegment

| name | mult | type | description |
| --- | --- | --- | --- |
| b0ch | 1..1 | [Susceptance](#UML87) | Zero sequence shunt (charging) susceptance, uniformly distributed, of the entire line section. |
| g0ch | 1..1 | [Conductance](#UML80) | Zero sequence shunt (charging) conductance, uniformly distributed, of the entire line section. |
| r0 | 1..1 | [Resistance](#UML86) | Zero sequence series resistance of the entire line section. |
| shortCircuitEndTemperature | 1..1 | [Temperature](#UML88) | 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. |
| x0 | 1..1 | [Reactance](#UML85) | Zero sequence series reactance of the entire line section. |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (Description) AsynchronousMachine

Inheritance path = [RotatingMachine](#UML2036) : [RegulatingCondEq](#UML2034) : [EnergyConnection](#UML2032) : [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

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.

Table 296 shows all attributes of AsynchronousMachine.

Table 296 – Attributes of LTDSShortCircuitProfile::AsynchronousMachine

| name | mult | type | description |
| --- | --- | --- | --- |
| converterFedDrive | 1..1 | [Boolean](#UML90) | Indicates whether the machine is a converter fed drive. Used for short circuit data exchange according to IEC 60909. |
| efficiency | 1..1 | [PerCent](#UML83) | Efficiency of the asynchronous machine at nominal operation as a percentage. Indicator for converter drive motors. Used for short circuit data exchange according to IEC 60909. |
| iaIrRatio | 1..1 | [Float](#UML93) | Ratio of locked-rotor current to the rated current of the motor (Ia/Ir). Used for short circuit data exchange according to IEC 60909. |
| polePairNumber | 1..1 | [Integer](#UML94) | Number of pole pairs of stator. Used for short circuit data exchange according to IEC 60909. |
| ratedMechanicalPower | 1..1 | [ActivePower](#UML75) | Rated mechanical power (Pr in IEC 60909-0). Used for short circuit data exchange according to IEC 60909. |
| reversible | 1..1 | [Boolean](#UML90) | Indicates for converter drive motors if the power can be reversible. Used for short circuit data exchange according to IEC 60909. |
| rxLockedRotorRatio | 0..1 | [Float](#UML93) | Locked rotor ratio (R/X). Used for short circuit data exchange according to IEC 60909. |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (Description) BusbarSection

Inheritance path = [Connector](#UML2027) : [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

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 voltage transformers 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.

Table 297 shows all attributes of BusbarSection.

Table 297 – Attributes of LTDSShortCircuitProfile::BusbarSection

| name | mult | type | description |
| --- | --- | --- | --- |
| ipMax | 0..1 | [CurrentFlow](#UML81) | Maximum allowable peak short-circuit current of busbar (Ipmax in IEC 60909-0).  Mechanical limit of the busbar in the substation itself. Used for short circuit data exchange according to IEC 60909. |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (abstract) ConductingEquipment

Inheritance path = [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

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

Table 298 shows all attributes of ConductingEquipment.

Table 298 – Attributes of LTDSShortCircuitProfile::ConductingEquipment

| name | mult | type | description |
| --- | --- | --- | --- |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (abstract) Conductor

Inheritance path = [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

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

Table 299 shows all attributes of Conductor.

Table 299 – Attributes of LTDSShortCircuitProfile::Conductor

| name | mult | type | description |
| --- | --- | --- | --- |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (abstract) Connector

Inheritance path = [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

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.

Table 300 shows all attributes of Connector.

Table 300 – Attributes of LTDSShortCircuitProfile::Connector

| name | mult | type | description |
| --- | --- | --- | --- |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (abstract) EarthFaultCompensator

Inheritance path = [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

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

Table 301 shows all attributes of EarthFaultCompensator.

Table 301 – Attributes of LTDSShortCircuitProfile::EarthFaultCompensator

| name | mult | type | description |
| --- | --- | --- | --- |
| r | 0..1 | [Resistance](#UML86) | Nominal resistance of device. |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (abstract) EnergyConnection

Inheritance path = [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

A connection of energy generation or consumption on the power system model.

Table 302 shows all attributes of EnergyConnection.

Table 302 – Attributes of LTDSShortCircuitProfile::EnergyConnection

| name | mult | type | description |
| --- | --- | --- | --- |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (Description) EnergySource

Inheritance path = [EnergyConnection](#UML2032) : [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

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

Table 303 shows all attributes of EnergySource.

Table 303 – Attributes of LTDSShortCircuitProfile::EnergySource

| name | mult | type | description |
| --- | --- | --- | --- |
| r | 0..1 | [Resistance](#UML86) | Positive sequence Thevenin resistance. |
| r0 | 0..1 | [Resistance](#UML86) | Zero sequence Thevenin resistance. |
| rn | 0..1 | [Resistance](#UML86) | Negative sequence Thevenin resistance. |
| x | 0..1 | [Reactance](#UML85) | Positive sequence Thevenin reactance. |
| x0 | 0..1 | [Reactance](#UML85) | Zero sequence Thevenin reactance. |
| xn | 0..1 | [Reactance](#UML85) | Negative sequence Thevenin reactance. |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (abstract) Equipment

Inheritance path = [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

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

Table 304 shows all attributes of Equipment.

Table 304 – Attributes of LTDSShortCircuitProfile::Equipment

| name | mult | type | description |
| --- | --- | --- | --- |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (Description) EquivalentBranch

Inheritance path = [EquivalentEquipment](#UML2041) : [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

The class represents equivalent branches. In cases where a transformer phase shift is modelled and the EquivalentBranch is spanning the same nodes, the impedance quantities for the EquivalentBranch shall consider the needed phase shift.

Table 305 shows all attributes of EquivalentBranch.

Table 305 – Attributes of LTDSShortCircuitProfile::EquivalentBranch

| name | mult | type | description |
| --- | --- | --- | --- |
| negativeR12 | 1..1 | [Resistance](#UML86) | 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 | 1..1 | [Resistance](#UML86) | 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 | 1..1 | [Reactance](#UML85) | 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 | 1..1 | [Reactance](#UML85) | 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 | 1..1 | [Resistance](#UML86) | 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 | 1..1 | [Resistance](#UML86) | 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 | 1..1 | [Reactance](#UML85) | 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 | 1..1 | [Reactance](#UML85) | 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. |
| zeroR12 | 1..1 | [Resistance](#UML86) | 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 | 1..1 | [Resistance](#UML86) | 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 | 1..1 | [Reactance](#UML85) | 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 | 1..1 | [Reactance](#UML85) | 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. |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (abstract) EquivalentEquipment

Inheritance path = [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

The class represents equivalent objects that are the result of a network reduction. The class is the base for equivalent objects of different types.

Table 306 shows all attributes of EquivalentEquipment.

Table 306 – Attributes of LTDSShortCircuitProfile::EquivalentEquipment

| name | mult | type | description |
| --- | --- | --- | --- |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (Description) EquivalentInjection

Inheritance path = [EquivalentEquipment](#UML2041) : [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

This class represents equivalent injections (generation or load). Voltage regulation is allowed only at the point of connection.

Table 307 shows all attributes of EquivalentInjection.

Table 307 – Attributes of LTDSShortCircuitProfile::EquivalentInjection

| name | mult | type | description |
| --- | --- | --- | --- |
| r | 1..1 | [Resistance](#UML86) | 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 | 1..1 | [Resistance](#UML86) | 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 | 1..1 | [Resistance](#UML86) | Negative sequence resistance. Used to represent Extended-Ward (IEC 60909).  Usage : Extended-Ward is a result of network reduction prior to the data exchange. |
| x | 1..1 | [Reactance](#UML85) | 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 | 1..1 | [Reactance](#UML85) | 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 | 1..1 | [Reactance](#UML85) | Negative sequence reactance. Used to represent Extended-Ward (IEC 60909).  Usage : Extended-Ward is a result of network reduction prior to the data exchange. |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (Description) ExternalNetworkInjection

Inheritance path = [RegulatingCondEq](#UML2034) : [EnergyConnection](#UML2032) : [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

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

Table 308 shows all attributes of ExternalNetworkInjection.

Table 308 – Attributes of LTDSShortCircuitProfile::ExternalNetworkInjection

| name | mult | type | description |
| --- | --- | --- | --- |
| ikSecond | 0..1 | [Boolean](#UML90) | Indicates whether initial symmetrical short-circuit current and power have been calculated according to IEC (Ik"). Used only if short circuit calculations are done according to superposition method. |
| maxInitialSymShCCurrent | 1..1 | [CurrentFlow](#UML81) | 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. |
| maxR0ToX0Ratio | 1..1 | [Float](#UML93) | 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 | 1..1 | [Float](#UML93) | 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 | 1..1 | [Float](#UML93) | 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 | 1..1 | [CurrentFlow](#UML81) | 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. |
| minR0ToX0Ratio | 1..1 | [Float](#UML93) | 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 | 1..1 | [Float](#UML93) | 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. |
| minZ0ToZ1Ratio | 1..1 | [Float](#UML93) | 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 | 0..1 | [PU](#UML84) | Voltage factor in pu, which was used to calculate short-circuit current Ik" and power Sk". Used only if short circuit calculations are done according to superposition method. |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (Description) GroundingImpedance

Inheritance path = [EarthFaultCompensator](#UML2029) : [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

A fixed impedance device used for grounding.

Table 309 shows all attributes of GroundingImpedance.

Table 309 – Attributes of LTDSShortCircuitProfile::GroundingImpedance

| name | mult | type | description |
| --- | --- | --- | --- |
| x | 1..1 | [Reactance](#UML85) | Reactance of device. |
| r | 0..1 | [Resistance](#UML86) | inherited from: [EarthFaultCompensator](#UML2029) |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (abstract) IdentifiedObject root class

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

Table 310 shows all attributes of IdentifiedObject.

Table 310 – Attributes of LTDSShortCircuitProfile::IdentifiedObject

| name | mult | type | description |
| --- | --- | --- | --- |
| mRID | 1..1 | [String](#UML95) | Master resource identifier issued by a model authority. The mRID is unique within an exchange context. Global uniqueness is easily achieved by using a UUID, as specified in RFC 4122, for the mRID. The use of UUID is strongly recommended.  For CIMXML data files in RDF syntax conforming to IEC 61970-552, the mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements. |

### (Description) LinearShuntCompensator

Inheritance path = [ShuntCompensator](#UML2039) : [RegulatingCondEq](#UML2034) : [EnergyConnection](#UML2032) : [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

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

Table 311 shows all attributes of LinearShuntCompensator.

Table 311 – Attributes of LTDSShortCircuitProfile::LinearShuntCompensator

| name | mult | type | description |
| --- | --- | --- | --- |
| b0PerSection | 1..1 | [Susceptance](#UML87) | Zero sequence shunt (charging) susceptance per section. |
| g0PerSection | 1..1 | [Conductance](#UML80) | Zero sequence shunt (charging) conductance per section. |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### MutualCoupling

Inheritance path = [IdentifiedObject](#UML68)

This class represents the zero sequence line mutual coupling.

Table 312 shows all attributes of MutualCoupling.

Table 312 – Attributes of LTDSShortCircuitProfile::MutualCoupling

| name | mult | type | description |
| --- | --- | --- | --- |
| b0ch | 1..1 | [Susceptance](#UML87) | Zero sequence mutual coupling shunt (charging) susceptance, uniformly distributed, of the entire line section. |
| distance11 | 1..1 | [Length](#UML82) | Distance to the start of the coupled region from the first line's terminal having sequence number equal to 1. |
| distance12 | 1..1 | [Length](#UML82) | Distance to the end of the coupled region from the first line's terminal with sequence number equal to 1. |
| distance21 | 1..1 | [Length](#UML82) | Distance to the start of coupled region from the second line's terminal with sequence number equal to 1. |
| distance22 | 1..1 | [Length](#UML82) | Distance to the end of coupled region from the second line's terminal with sequence number equal to 1. |
| g0ch | 1..1 | [Conductance](#UML80) | Zero sequence mutual coupling shunt (charging) conductance, uniformly distributed, of the entire line section. |
| r0 | 1..1 | [Resistance](#UML86) | Zero sequence branch-to-branch mutual impedance coupling, resistance. |
| x0 | 1..1 | [Reactance](#UML85) | Zero sequence branch-to-branch mutual impedance coupling, reactance. |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

Table 313 shows all association ends of MutualCoupling with other classes.

Table 313 – Association ends of LTDSShortCircuitProfile::MutualCoupling with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Second\_Terminal | 1..1 | [Terminal](#UML2020) | The starting terminal for the calculation of distances along the second branch of the mutual coupling. |
| 0..\* | First\_Terminal | 1..1 | [Terminal](#UML2020) | 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. |

### (Description) NonlinearShuntCompensatorPoint root class

A non linear shunt compensator bank or section admittance value. The number of NonlinearShuntCompenstorPoint instances associated with a NonlinearShuntCompensator shall be equal to ShuntCompensator.maximumSections. ShuntCompensator.sections shall only be set to one of the NonlinearShuntCompenstorPoint.sectionNumber. There is no interpolation between NonlinearShuntCompenstorPoint-s.

Table 314 shows all attributes of NonlinearShuntCompensatorPoint.

Table 314 – Attributes of LTDSShortCircuitProfile::NonlinearShuntCompensatorPoint

| name | mult | type | description |
| --- | --- | --- | --- |
| b0 | 1..1 | [Susceptance](#UML87) | Zero sequence shunt (charging) susceptance per section. |
| g0 | 1..1 | [Conductance](#UML80) | Zero sequence shunt (charging) conductance per section. |

### (Description) PetersenCoil

Inheritance path = [EarthFaultCompensator](#UML2029) : [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

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

Table 315 shows all attributes of PetersenCoil.

Table 315 – Attributes of LTDSShortCircuitProfile::PetersenCoil

| name | mult | type | description |
| --- | --- | --- | --- |
| mode | 1..1 | [PetersenCoilModeKind](#UML70) | The mode of operation of the Petersen coil. |
| nominalU | 1..1 | [Voltage](#UML89) | The nominal voltage for which the coil is designed. |
| offsetCurrent | 0..1 | [CurrentFlow](#UML81) | 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 A depending on voltage and resonance conditions. |
| positionCurrent | 0..1 | [CurrentFlow](#UML81) | The control current used to control the Petersen coil also known as the position current. Typically in the range of 20 mA to 200 mA. |
| xGroundMax | 1..1 | [Reactance](#UML85) | The maximum reactance. |
| xGroundMin | 1..1 | [Reactance](#UML85) | The minimum reactance. |
| xGroundNominal | 1..1 | [Reactance](#UML85) | 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](#UML86) | inherited from: [EarthFaultCompensator](#UML2029) |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (abstract) PowerSystemResource

Inheritance path = [IdentifiedObject](#UML68)

A power system resource (PSR) 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.

Table 316 shows all attributes of PowerSystemResource.

Table 316 – Attributes of LTDSShortCircuitProfile::PowerSystemResource

| name | mult | type | description |
| --- | --- | --- | --- |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (Description) PowerTransformer

Inheritance path = [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

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 modelled 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.

Table 317 shows all attributes of PowerTransformer.

Table 317 – Attributes of LTDSShortCircuitProfile::PowerTransformer

| name | mult | type | description |
| --- | --- | --- | --- |
| beforeShCircuitHighestOperatingCurrent | 0..1 | [CurrentFlow](#UML81) | The highest operating current (Ib in 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 | 0..1 | [Voltage](#UML89) | The highest operating voltage (Ub in IEC 60909-0) before short circuit. It is used for calculation of the impedance correction factor KT defined in IEC 60909-0. This is worst case voltage on the low side winding (3.7.1 of IEC 60909:2001). Used to define operating conditions. |
| beforeShortCircuitAnglePf | 0..1 | [AngleDegrees](#UML79) | The angle of power factor before short circuit (phib in 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 | 0..1 | [Voltage](#UML89) | The minimum operating voltage (uQmin in 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 | 1..1 | [Boolean](#UML90) | Indicates whether the machine is part of a power station unit. Used for short circuit data exchange according to IEC 60909. It has an impact on how the correction factors are calculated for transformers, since the transformer is not necessarily part of a synchronous machine and generating unit. It is not always possible to derive this information from the model. This is why the attribute is necessary. |
| operationalValuesConsidered | 0..1 | [Boolean](#UML90) | 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. |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (Description) PowerTransformerEnd

Inheritance path = [TransformerEnd](#UML2046) : [IdentifiedObject](#UML68)

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 (TransformerEnd.endNumber=1) PowerTransformerEnd has non zero values on r, r0, x, and x0 while the low voltage (TransformerEnd.endNumber=2) PowerTransformerEnd has zero values for r, r0, x, and x0. Parameters are always provided, even if the PowerTransformerEnds have the same rated voltage. In this case, the parameters are provided at the PowerTransformerEnd which has TransformerEnd.endNumber equal to 1.

2) for a three Terminal PowerTransformer the three PowerTransformerEnds represent a star equivalent with each leg in the star represented by r, r0, x, and x0 values.

3) For a three Terminal transformer each PowerTransformerEnd shall have g, g0, b and b0 values corresponding to the no load losses distributed on the three PowerTransformerEnds. The total no load loss shunt impedances may also be placed at one of the PowerTransformerEnds, preferably the end numbered 1, having the shunt values on end 1. This is the preferred way.

4) 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).

Table 318 shows all attributes of PowerTransformerEnd.

Table 318 – Attributes of LTDSShortCircuitProfile::PowerTransformerEnd

| name | mult | type | description |
| --- | --- | --- | --- |
| b0 | 1..1 | [Susceptance](#UML87) | Zero sequence magnetizing branch susceptance. |
| phaseAngleClock | 1..1 | [Integer](#UML94) | 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. |
| g0 | 1..1 | [Conductance](#UML80) | Zero sequence magnetizing branch conductance (star-model). |
| r0 | 1..1 | [Resistance](#UML86) | Zero sequence series resistance (star-model) of the transformer end. |
| x0 | 1..1 | [Reactance](#UML85) | Zero sequence series reactance of the transformer end. |
| rground | 0..1 | [Resistance](#UML86) | inherited from: [TransformerEnd](#UML2046) |
| grounded | 1..1 | [Boolean](#UML90) | inherited from: [TransformerEnd](#UML2046) |
| xground | 0..1 | [Reactance](#UML85) | inherited from: [TransformerEnd](#UML2046) |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (abstract) RegulatingCondEq

Inheritance path = [EnergyConnection](#UML2032) : [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

A type of conducting equipment that can regulate a quantity (i.e. voltage or flow) at a specific point in the network.

Table 319 shows all attributes of RegulatingCondEq.

Table 319 – Attributes of LTDSShortCircuitProfile::RegulatingCondEq

| name | mult | type | description |
| --- | --- | --- | --- |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (abstract) RotatingMachine

Inheritance path = [RegulatingCondEq](#UML2034) : [EnergyConnection](#UML2032) : [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

A rotating machine which may be used as a generator or motor.

Table 320 shows all attributes of RotatingMachine.

Table 320 – Attributes of LTDSShortCircuitProfile::RotatingMachine

| name | mult | type | description |
| --- | --- | --- | --- |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (Description) SeriesCompensator

Inheritance path = [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

A Series Compensator is a series capacitor or reactor or an AC transmission line without charging susceptance. It is a two terminal device.

Table 321 shows all attributes of SeriesCompensator.

Table 321 – Attributes of LTDSShortCircuitProfile::SeriesCompensator

| name | mult | type | description |
| --- | --- | --- | --- |
| r0 | 1..1 | [Resistance](#UML86) | Zero sequence resistance. |
| x0 | 1..1 | [Reactance](#UML85) | Zero sequence reactance. |
| varistorPresent | 1..1 | [Boolean](#UML90) | Describe if a metal oxide varistor (mov) for over voltage protection is configured in parallel with the series compensator. It is used for short circuit calculations. |
| varistorRatedCurrent | 0..1 | [CurrentFlow](#UML81) | The maximum current the varistor is designed to handle at specified duration. It is used for short circuit calculations and exchanged only if SeriesCompensator.varistorPresent is true.  The attribute shall be a positive value. |
| varistorVoltageThreshold | 0..1 | [Voltage](#UML89) | The dc voltage at which the varistor starts conducting. It is used for short circuit calculations and exchanged only if SeriesCompensator.varistorPresent is true. |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (abstract) ShuntCompensator

Inheritance path = [RegulatingCondEq](#UML2034) : [EnergyConnection](#UML2032) : [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

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 bPerSection indicates that the compensator is a reactor. ShuntCompensator is a single terminal device. Ground is implied.

Table 322 shows all attributes of ShuntCompensator.

Table 322 – Attributes of LTDSShortCircuitProfile::ShuntCompensator

| name | mult | type | description |
| --- | --- | --- | --- |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (Description) SynchronousMachine

Inheritance path = [RotatingMachine](#UML2036) : [RegulatingCondEq](#UML2034) : [EnergyConnection](#UML2032) : [ConductingEquipment](#UML2024) : [Equipment](#UML2023) : [PowerSystemResource](#UML2022) : [IdentifiedObject](#UML68)

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.

Table 323 shows all attributes of SynchronousMachine.

Table 323 – Attributes of LTDSShortCircuitProfile::SynchronousMachine

| name | mult | type | description |
| --- | --- | --- | --- |
| earthing | 1..1 | [Boolean](#UML90) | Indicates whether or not the generator is earthed. Used for short circuit data exchange according to IEC 60909. |
| earthingStarPointR | 0..1 | [Resistance](#UML86) | Generator star point earthing resistance (Re). Used for short circuit data exchange according to IEC 60909. |
| earthingStarPointX | 0..1 | [Reactance](#UML85) | Generator star point earthing reactance (Xe). Used for short circuit data exchange according to IEC 60909. |
| ikk | 0..1 | [CurrentFlow](#UML81) | 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 in IEC 60909-0:2001).  Used only for single fed short circuit on a generator. (4.3.4.2. in IEC 60909-0:2001). |
| mu | 0..1 | [Float](#UML93) | Factor to calculate the breaking current (Section 4.5.2.1 in IEC 60909-0).  Used only for single fed short circuit on a generator (Section 4.3.4.2. in IEC 60909-0). |
| x0 | 1..1 | [Reactance](#UML85) | Zero sequence reactance of the synchronous machine. |
| r0 | 1..1 | [Resistance](#UML86) | Zero sequence resistance of the synchronous machine. |
| x2 | 1..1 | [Reactance](#UML85) | Negative sequence reactance. |
| r2 | 1..1 | [Resistance](#UML86) | Negative sequence resistance. |
| r | 1..1 | [Resistance](#UML86) | 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. |
| satDirectSubtransX | 1..1 | [PU](#UML84) | Direct-axis subtransient reactance saturated, also known as Xd"sat. |
| satDirectSyncX | 0..1 | [PU](#UML84) | 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. in IEC 60909-0:2001). |
| satDirectTransX | 0..1 | [PU](#UML84) | Saturated Direct-axis transient reactance. The attribute is primarily used for short circuit calculations according to ANSI. |
| shortCircuitRotorType | 0..1 | [ShortCircuitRotorKind](#UML71) | Type of rotor, used by short circuit applications, only for single fed short circuit according to IEC 60909. |
| voltageRegulationRange | 0..1 | [PerCent](#UML83) | Range of generator voltage regulation (PG in 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. |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (abstract) Terminal

Inheritance path = [ACDCTerminal](#UML2019) : [IdentifiedObject](#UML68)

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

Table 324 shows all attributes of Terminal.

Table 324 – Attributes of LTDSShortCircuitProfile::Terminal

| name | mult | type | description |
| --- | --- | --- | --- |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### (abstract) TransformerEnd

Inheritance path = [IdentifiedObject](#UML68)

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.

Table 325 shows all attributes of TransformerEnd.

Table 325 – Attributes of LTDSShortCircuitProfile::TransformerEnd

| name | mult | type | description |
| --- | --- | --- | --- |
| rground | 0..1 | [Resistance](#UML86) | (for Yn and Zn connections) Resistance part of neutral impedance where 'grounded' is true. |
| grounded | 1..1 | [Boolean](#UML90) | (for Yn and Zn connections) True if the neutral is solidly grounded. |
| xground | 0..1 | [Reactance](#UML85) | (for Yn and Zn connections) Reactive part of neutral impedance where 'grounded' is true. |
| mRID | 1..1 | [String](#UML95) | inherited from: [IdentifiedObject](#UML68) |

### PetersenCoilModeKind enumeration

The mode of operation for a Petersen coil.

Table 326 shows all literals of PetersenCoilModeKind.

Table 326 – Literals of LTDSShortCircuitProfile::PetersenCoilModeKind

| literal | value | description |
| --- | --- | --- |
| fixed |  | Fixed position. |
| manual |  | Manual positioning. |
| automaticPositioning |  | Automatic positioning. |

### ShortCircuitRotorKind enumeration

Type of rotor, used by short circuit applications.

Table 327 shows all literals of ShortCircuitRotorKind.

Table 327 – Literals of LTDSShortCircuitProfile::ShortCircuitRotorKind

| literal | value | description |
| --- | --- | --- |
| 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. |

### UnitMultiplier enumeration

The unit multipliers defined for the CIM. When applied to unit symbols, the unit symbol is treated as a derived unit. Regardless of the contents of the unit symbol text, the unit symbol shall be treated as if it were a single-character unit symbol. Unit symbols should not contain multipliers, and it should be left to the multiplier to define the multiple for an entire data type.

For example, if a unit symbol is "m2Pers" and the multiplier is "k", then the value is k(m\*\*2/s), and the multiplier applies to the entire final value, not to any individual part of the value. This can be conceptualized by substituting a derived unit symbol for the unit type. If one imagines that the symbol "Þ" represents the derived unit "m2Pers", then applying the multiplier "k" can be conceptualized simply as "kÞ".

For example, the SI unit for mass is "kg" and not "g". If the unit symbol is defined as "kg", then the multiplier is applied to "kg" as a whole and does not replace the "k" in front of the "g". In this case, the multiplier of "m" would be used with the unit symbol of "kg" to represent one gram. As a text string, this violates the instructions in IEC 80000-1. However, because the unit symbol in CIM is treated as a derived unit instead of as an SI unit, it makes more sense to conceptualize the "kg" as if it were replaced by one of the proposed replacements for the SI mass symbol. If one imagines that the "kg" were replaced by a symbol "Þ", then it is easier to conceptualize the multiplier "m" as creating the proper unit "mÞ", and not the forbidden unit "mkg".

Table 328 shows all literals of UnitMultiplier.

Table 328 – Literals of LTDSShortCircuitProfile::UnitMultiplier

| literal | value | description |
| --- | --- | --- |
| y | -24 | Yocto 10\*\*-24. |
| z | -21 | Zepto 10\*\*-21. |
| a | -18 | Atto 10\*\*-18. |
| f | -15 | Femto 10\*\*-15. |
| p | -12 | Pico 10\*\*-12. |
| n | -9 | Nano 10\*\*-9. |
| micro | -6 | Micro 10\*\*-6. |
| m | -3 | Milli 10\*\*-3. |
| c | -2 | Centi 10\*\*-2. |
| d | -1 | Deci 10\*\*-1. |
| none | 0 | No multiplier or equivalently multiply by 1. |
| da | 1 | Deca 10\*\*1. |
| h | 2 | Hecto 10\*\*2. |
| k | 3 | Kilo 10\*\*3. |
| M | 6 | Mega 10\*\*6. |
| G | 9 | Giga 10\*\*9. |
| T | 12 | Tera 10\*\*12. |
| P | 15 | Peta 10\*\*15. |
| E | 18 | Exa 10\*\*18. |
| Z | 21 | Zetta 10\*\*21. |
| Y | 24 | Yotta 10\*\*24. |

### UnitSymbol enumeration

The derived units defined for usage in the CIM. In some cases, the derived unit is equal to an SI unit. Whenever possible, the standard derived symbol is used instead of the formula for the derived unit. For example, the unit symbol Farad is defined as "F" instead of "CPerV". In cases where a standard symbol does not exist for a derived unit, the formula for the unit is used as the unit symbol. For example, density does not have a standard symbol and so it is represented as "kgPerm3". With the exception of the "kg", which is an SI unit, the unit symbols do not contain multipliers and therefore represent the base derived unit to which a multiplier can be applied as a whole.

Every unit symbol is treated as an unparseable text as if it were a single-letter symbol. The meaning of each unit symbol is defined by the accompanying descriptive text and not by the text contents of the unit symbol.

To allow the widest possible range of serializations without requiring special character handling, several substitutions are made which deviate from the format described in IEC 80000-1. The division symbol "/" is replaced by the letters "Per". Exponents are written in plain text after the unit as "m3" instead of being formatted as "m" with a superscript of 3 or introducing a symbol as in "m^3". The degree symbol "°" is replaced with the letters "deg". Any clarification of the meaning for a substitution is included in the description for the unit symbol.

Non-SI units are included in list of unit symbols to allow sources of data to be correctly labelled with their non-SI units (for example, a GPS sensor that is reporting numbers that represent feet instead of meters). This allows software to use the unit symbol information correctly convert and scale the raw data of those sources into SI-based units.

The integer values are used for harmonization with IEC 61850.

Table 329 shows all literals of UnitSymbol.

Table 329 – Literals of LTDSShortCircuitProfile::UnitSymbol

| literal | value | description |
| --- | --- | --- |
| none | 0 | Dimension less quantity, e.g. count, per unit, etc. |
| m | 2 | Length in metres. |
| kg | 3 | Mass in kilograms. Note: multiplier “k” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| s | 4 | Time in seconds. |
| A | 5 | Current in amperes. |
| K | 6 | Temperature in kelvins. |
| mol | 7 | Amount of substance in moles. |
| cd | 8 | Luminous intensity in candelas. |
| deg | 9 | Plane angle in degrees. |
| rad | 10 | Plane angle in radians (m/m). |
| sr | 11 | Solid angle in steradians (m2/m2). |
| Gy | 21 | Absorbed dose in grays (J/kg). |
| Bq | 22 | Radioactivity in becquerels (1/s). |
| degC | 23 | 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 from coulomb the symbol used in the UML is degC. The reason for not using °C is that the special character ° is difficult to manage in software. |
| Sv | 24 | Dose equivalent in sieverts (J/kg). |
| F | 25 | Electric capacitance in farads (C/V). |
| C | 26 | Electric charge in coulombs (A·s). |
| S | 27 | Conductance in siemens. |
| H | 28 | Electric inductance in henrys (Wb/A). |
| V | 29 | Electric potential in volts (W/A). |
| ohm | 30 | Electric resistance in ohms (V/A). |
| J | 31 | Energy in joules (N·m = C·V = W·s). |
| N | 32 | Force in newtons (kg·m/s²). |
| Hz | 33 | Frequency in hertz (1/s). |
| lx | 34 | Illuminance in lux (lm/m²). |
| lm | 35 | Luminous flux in lumens (cd·sr). |
| Wb | 36 | Magnetic flux in webers (V·s). |
| T | 37 | Magnetic flux density in teslas (Wb/m2). |
| W | 38 | Real power in watts (J/s). Electrical power may have real and reactive components. The real portion of electrical power (I²R or VIcos(phi)), is expressed in Watts. See also apparent power and reactive power. |
| Pa | 39 | Pressure in pascals (N/m²). Note: the absolute or relative measurement of pressure is implied with this entry. See below for more explicit forms. |
| m2 | 41 | Area in square metres (m²). |
| m3 | 42 | Volume in cubic metres (m³). |
| mPers | 43 | Velocity in metres per second (m/s). |
| mPers2 | 44 | Acceleration in metres per second squared (m/s²). |
| m3Pers | 45 | Volumetric flow rate in cubic metres per second (m³/s). |
| mPerm3 | 46 | Fuel efficiency in metres per cubic metres (m/m³). |
| kgm | 47 | Moment of mass in kilogram metres (kg·m) (first moment of mass). Note: multiplier “k” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| kgPerm3 | 48 | Density in kilogram/cubic metres (kg/m³). Note: multiplier “k” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| m2Pers | 49 | Viscosity in square metres / second (m²/s). |
| WPermK | 50 | Thermal conductivity in watt/metres kelvin. |
| JPerK | 51 | Heat capacity in joules/kelvin. |
| ppm | 52 | Concentration in parts per million. |
| rotPers | 53 | Rotations per second (1/s). See also Hz (1/s). |
| radPers | 54 | Angular velocity in radians per second (rad/s). |
| WPerm2 | 55 | Heat flux density, irradiance, watts per square metre. |
| JPerm2 | 56 | Insulation energy density, joules per square metre or watt second per square metre. |
| SPerm | 57 | Conductance per length (F/m). |
| KPers | 58 | Temperature change rate in kelvins per second. |
| PaPers | 59 | Pressure change rate in pascals per second. |
| JPerkgK | 60 | Specific heat capacity, specific entropy, joules per kilogram Kelvin. |
| VA | 61 | Apparent power in volt amperes. See also real power and reactive power. |
| VAr | 63 | Reactive power in volt amperes reactive. The “reactive” or “imaginary” component of electrical power (VIsin(phi)). (See also real power and apparent power).  Note: Different meter designs use different methods to arrive at their results. Some meters may compute reactive power as an arithmetic value, while others compute the value vectorially. The data consumer should determine the method in use and the suitability of the measurement for the intended purpose. |
| cosPhi | 65 | Power factor, dimensionless.  Note 1: This definition of power factor only holds for balanced systems. See the alternative definition under code 153.  Note 2 : Beware of differing sign conventions in use between the IEC and EEI. It is assumed that the data consumer understands the type of meter in use and the sign convention in use by the utility. |
| Vs | 66 | Volt seconds (Ws/A). |
| V2 | 67 | Volt squared (W²/A²). |
| As | 68 | Ampere seconds (A·s). |
| A2 | 69 | Amperes squared (A²). |
| A2s | 70 | Ampere squared time in square amperes (A²s). |
| VAh | 71 | Apparent energy in volt ampere hours. |
| Wh | 72 | Real energy in watt hours. |
| VArh | 73 | Reactive energy in volt ampere reactive hours. |
| VPerHz | 74 | Magnetic flux in volt per hertz. |
| HzPers | 75 | Rate of change of frequency in hertz per second. |
| character | 76 | Number of characters. |
| charPers | 77 | Data rate (baud) in characters per second. |
| kgm2 | 78 | Moment of mass in kilogram square metres (kg·m²) (Second moment of mass, commonly called the moment of inertia). Note: multiplier “k” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| dB | 79 | Sound pressure level in decibels. Note: multiplier “d” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| WPers | 81 | Ramp rate in watts per second. |
| lPers | 82 | Volumetric flow rate in litres per second. |
| dBm | 83 | Power level (logarithmic ratio of signal strength , Bel-mW), normalized to 1mW. Note: multiplier “d” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| h | 84 | Time in hours, hour = 60 min = 3600 s. |
| min | 85 | Time in minutes, minute = 60 s. |
| Q | 100 | Quantity power, Q. |
| Qh | 101 | Quantity energy, Qh. |
| ohmm | 102 | Resistivity, ohm metres, (rho). |
| APerm | 103 | A/m, magnetic field strength, amperes per metre. |
| V2h | 104 | Volt-squared hour, volt-squared-hours. |
| A2h | 105 | Ampere-squared hour, ampere-squared hour. |
| Ah | 106 | Ampere-hours, ampere-hours. |
| count | 111 | Amount of substance, Counter value. |
| ft3 | 119 | Volume, cubic feet. |
| m3Perh | 125 | Volumetric flow rate, cubic metres per hour. |
| gal | 128 | Volume in gallons, US gallon (1 gal = 231 in3 = 128 fl ounce). |
| Btu | 132 | Energy, British Thermal Units. |
| l | 134 | Volume in litres, litre = dm3 = m3/1000. |
| lPerh | 137 | Volumetric flow rate, litres per hour. |
| lPerl | 143 | Concentration, The ratio of the volume of a solute divided by the volume of the solution. Note: Users may need use a prefix such a ‘µ’ to express a quantity such as ‘µL/L’. |
| gPerg | 144 | Concentration, The ratio of the mass of a solute divided by the mass of the solution. Note: Users may need use a prefix such a ‘µ’ to express a quantity such as ‘µg/g’. |
| molPerm3 | 145 | Concentration, The amount of substance concentration, (c), the amount of solvent in moles divided by the volume of solution in m³. |
| molPermol | 146 | Concentration, Molar fraction, the ratio of the molar amount of a solute divided by the molar amount of the solution. |
| molPerkg | 147 | Concentration, Molality, the amount of solute in moles and the amount of solvent in kilograms. |
| sPers | 149 | Time, Ratio of time. Note: Users may need to supply a prefix such as ‘µ’ to show rates such as ‘µs/s’. |
| HzPerHz | 150 | Frequency, rate of frequency change. Note: Users may need to supply a prefix such as ‘m’ to show rates such as ‘mHz/Hz’. |
| VPerV | 151 | Voltage, ratio of voltages. Note: Users may need to supply a prefix such as ‘m’ to show rates such as ‘mV/V’. |
| APerA | 152 | Current, ratio of amperages. Note: Users may need to supply a prefix such as ‘m’ to show rates such as ‘mA/A’. |
| VPerVA | 153 | Power factor, PF, the ratio of the active power to the apparent power. Note: The sign convention used for power factor will differ between IEC meters and EEI (ANSI) meters. It is assumed that the data consumers understand the type of meter being used and agree on the sign convention in use at any given utility. |
| rev | 154 | Amount of rotation, revolutions. |
| kat | 158 | Catalytic activity, katal = mol / s. |
| JPerkg | 165 | Specific energy, Joules / kg. |
| m3Uncompensated | 166 | Volume, cubic metres, with the value uncompensated for weather effects. |
| m3Compensated | 167 | Volume, cubic metres, with the value compensated for weather effects. |
| WPerW | 168 | Signal Strength, ratio of power. Note: Users may need to supply a prefix such as ‘m’ to show rates such as ‘mW/W’. |
| therm | 169 | Energy, therms. |
| onePerm | 173 | Wavenumber, reciprocal metres, (1/m). |
| m3Perkg | 174 | Specific volume, cubic metres per kilogram, v. |
| Pas | 175 | Dynamic viscosity, pascal seconds. |
| Nm | 176 | Moment of force, newton metres. |
| NPerm | 177 | Surface tension, newton per metre. |
| radPers2 | 178 | Angular acceleration, radians per second squared. |
| JPerm3 | 181 | Energy density, joules per cubic metre. |
| VPerm | 182 | Electric field strength, volts per metre. |
| CPerm3 | 183 | Electric charge density, coulombs per cubic metre. |
| CPerm2 | 184 | Surface charge density, coulombs per square metre. |
| FPerm | 185 | Permittivity, farads per metre. |
| HPerm | 186 | Permeability, henrys per metre. |
| JPermol | 187 | Molar energy, joules per mole. |
| JPermolK | 188 | Molar entropy, molar heat capacity, joules per mole kelvin. |
| CPerkg | 189 | Exposure (x rays), coulombs per kilogram. |
| GyPers | 190 | Absorbed dose rate, grays per second. |
| WPersr | 191 | Radiant intensity, watts per steradian. |
| WPerm2sr | 192 | Radiance, watts per square metre steradian. |
| katPerm3 | 193 | Catalytic activity concentration, katals per cubic metre. |
| d | 195 | Time in days, day = 24 h = 86400 s. |
| anglemin | 196 | Plane angle, minutes. |
| anglesec | 197 | Plane angle, seconds. |
| ha | 198 | Area, hectares. |
| tonne | 199 | Mass in tons, “tonne” or “metric ton” (1000 kg = 1 Mg). |
| bar | 214 | Pressure in bars, (1 bar = 100 kPa). |
| mmHg | 215 | Pressure, millimetres of mercury (1 mmHg is approximately 133.3 Pa). |
| M | 217 | Length, nautical miles (1 M = 1852 m). |
| kn | 219 | Speed, knots (1 kn = 1852/3600) m/s. |
| Mx | 276 | Magnetic flux, maxwells (1 Mx = 10-8 Wb). |
| G | 277 | Magnetic flux density, gausses (1 G = 10-4 T). |
| Oe | 278 | Magnetic field in oersteds, (1 Oe = (103/4p) A/m). |
| Vh | 280 | Volt-hour, Volt hours. |
| WPerA |  | Active power per current flow, watts per Ampere. |
| onePerHz |  | Reciprocal of frequency (1/Hz). |
| VPerVAr |  | Power factor, PF, the ratio of the active power to the apparent power. Note: The sign convention used for power factor will differ between IEC meters and EEI (ANSI) meters. It is assumed that the data consumers understand the type of meter being used and agree on the sign convention in use at any given utility. |
| ohmPerm | 86 | Electric resistance per length in ohms per metre ((V/A)/m). |
| kgPerJ |  | Weight per energy in kilograms per joule (kg/J). Note: multiplier “k” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| JPers |  | Energy rate in joules per second (J/s). |

### ReactancePerLength datatype

Reactance (imaginary part of impedance) per unit of length, at rated frequency.

Table 330 shows all attributes of ReactancePerLength.

Table 330 – Attributes of LTDSShortCircuitProfile::ReactancePerLength

| name | mult | type | description |
| --- | --- | --- | --- |
| multiplier | 0..1 | [UnitMultiplier](#UML72) | (const=none) |
| unit | 0..1 | [UnitSymbol](#UML73) | (const=ohmPerm) |
| value | 0..1 | [Float](#UML93) |  |

### ActivePower datatype

Product of RMS value of the voltage and the RMS value of the in-phase component of the current.

Table 331 shows all attributes of ActivePower.

Table 331 – Attributes of LTDSShortCircuitProfile::ActivePower

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML93) |  |
| multiplier | 0..1 | [UnitMultiplier](#UML72) | (const=M) |
| unit | 0..1 | [UnitSymbol](#UML73) | (const=W) |

### ResistancePerLength datatype

Resistance (real part of impedance) per unit of length.

Table 332 shows all attributes of ResistancePerLength.

Table 332 – Attributes of LTDSShortCircuitProfile::ResistancePerLength

| name | mult | type | description |
| --- | --- | --- | --- |
| multiplier | 0..1 | [UnitMultiplier](#UML72) | (const=none) |
| unit | 0..1 | [UnitSymbol](#UML73) | (const=ohmPerm) |
| value | 0..1 | [Float](#UML93) |  |

### SusceptancePerLength datatype

Imaginary part of admittance per unit of length.

Table 333 shows all attributes of SusceptancePerLength.

Table 333 – Attributes of LTDSShortCircuitProfile::SusceptancePerLength

| name | mult | type | description |
| --- | --- | --- | --- |
| multiplier | 0..1 | [UnitMultiplier](#UML72) | (const=none) |
| unit | 0..1 | [UnitSymbol](#UML73) | (const=SPerm) |
| value | 0..1 | [Float](#UML93) |  |

### ConductancePerLength datatype

Real part of admittance per unit of length.

Table 334 shows all attributes of ConductancePerLength.

Table 334 – Attributes of LTDSShortCircuitProfile::ConductancePerLength

| name | mult | type | description |
| --- | --- | --- | --- |
| multiplier | 0..1 | [UnitMultiplier](#UML72) | (const=none) |
| unit | 0..1 | [UnitSymbol](#UML73) | (const=SPerm) |
| value | 0..1 | [Float](#UML93) |  |

### AngleDegrees datatype

Measurement of angle in degrees.

Table 335 shows all attributes of AngleDegrees.

Table 335 – Attributes of LTDSShortCircuitProfile::AngleDegrees

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML93) |  |
| unit | 0..1 | [UnitSymbol](#UML73) | (const=deg) |
| multiplier | 0..1 | [UnitMultiplier](#UML72) | (const=none) |

### Conductance datatype

Factor by which voltage must be multiplied to give corresponding power lost from a circuit. Real part of admittance.

Table 336 shows all attributes of Conductance.

Table 336 – Attributes of LTDSShortCircuitProfile::Conductance

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML93) |  |
| unit | 0..1 | [UnitSymbol](#UML73) | (const=S) |
| multiplier | 0..1 | [UnitMultiplier](#UML72) | (const=none) |

### CurrentFlow datatype

Electrical current with sign convention: positive flow is out of the conducting equipment into the connectivity node. Can be both AC and DC.

Table 337 shows all attributes of CurrentFlow.

Table 337 – Attributes of LTDSShortCircuitProfile::CurrentFlow

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML93) |  |
| multiplier | 0..1 | [UnitMultiplier](#UML72) | (const=none) |
| unit | 0..1 | [UnitSymbol](#UML73) | (const=A) |

### Length datatype

Unit of length. It shall be a positive value or zero.

Table 338 shows all attributes of Length.

Table 338 – Attributes of LTDSShortCircuitProfile::Length

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML93) |  |
| unit | 0..1 | [UnitSymbol](#UML73) | (const=m) |
| multiplier | 0..1 | [UnitMultiplier](#UML72) | (const=k) |

### PerCent datatype

Percentage on a defined base. For example, specify as 100 to indicate at the defined base.

Table 339 shows all attributes of PerCent.

Table 339 – Attributes of LTDSShortCircuitProfile::PerCent

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML93) | Normally 0 to 100 on a defined base. |
| unit | 0..1 | [UnitSymbol](#UML73) | (const=none) |
| multiplier | 0..1 | [UnitMultiplier](#UML72) | (const=none) |

### PU datatype

Per Unit - a positive or negative value referred to a defined base. Values typically range from -10 to +10.

Table 340 shows all attributes of PU.

Table 340 – Attributes of LTDSShortCircuitProfile::PU

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML93) |  |
| unit | 0..1 | [UnitSymbol](#UML73) | (const=none) |
| multiplier | 0..1 | [UnitMultiplier](#UML72) | (const=none) |

### Reactance datatype

Reactance (imaginary part of impedance), at rated frequency.

Table 341 shows all attributes of Reactance.

Table 341 – Attributes of LTDSShortCircuitProfile::Reactance

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML93) |  |
| unit | 0..1 | [UnitSymbol](#UML73) | (const=ohm) |
| multiplier | 0..1 | [UnitMultiplier](#UML72) | (const=none) |

### Resistance datatype

Resistance (real part of impedance).

Table 342 shows all attributes of Resistance.

Table 342 – Attributes of LTDSShortCircuitProfile::Resistance

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML93) |  |
| unit | 0..1 | [UnitSymbol](#UML73) | (const=ohm) |
| multiplier | 0..1 | [UnitMultiplier](#UML72) | (const=none) |

### Susceptance datatype

Imaginary part of admittance.

Table 343 shows all attributes of Susceptance.

Table 343 – Attributes of LTDSShortCircuitProfile::Susceptance

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML93) |  |
| unit | 0..1 | [UnitSymbol](#UML73) | (const=S) |
| multiplier | 0..1 | [UnitMultiplier](#UML72) | (const=none) |

### Temperature datatype

Value of temperature in degrees Celsius.

Table 344 shows all attributes of Temperature.

Table 344 – Attributes of LTDSShortCircuitProfile::Temperature

| name | mult | type | description |
| --- | --- | --- | --- |
| multiplier | 0..1 | [UnitMultiplier](#UML72) | (const=none) |
| unit | 0..1 | [UnitSymbol](#UML73) | (const=degC) |
| value | 0..1 | [Float](#UML93) |  |

### Voltage datatype

Electrical voltage, can be both AC and DC.

Table 345 shows all attributes of Voltage.

Table 345 – Attributes of LTDSShortCircuitProfile::Voltage

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML93) |  |
| multiplier | 0..1 | [UnitMultiplier](#UML72) | (const=k) |
| unit | 0..1 | [UnitSymbol](#UML73) | (const=V) |

### Boolean primitive

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

### Date primitive

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

### DateTime primitive

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

### Float primitive

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

### Integer primitive

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

### String primitive

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

### (profcim) IRI primitive

An IRI (Internationalized Resource Identifier) within an RDF graph is a Unicode string that conforms to the syntax defined in RFC 3987.

IRIs in the RDF abstract syntax must be absolute, and may contain a fragment identifier.

IRI equality: Two IRIs are equal if and only if they are equivalent under Simple String Comparison according to section 5.1 of [RFC3987]. Further normalization must not be performed when comparing IRIs for equality.

IRIs are a generalization of URIs [RFC3986] that permits a wider range of Unicode characters. Every absolute URI and URL is an IRI, but not every IRI is an URI. When IRIs are used in operations that are only defined for URIs, they must first be converted according to the mapping defined in section 3.1 of [RFC3987]. A notable example is retrieval over the HTTP protocol. The mapping involves UTF-8 encoding of non-ASCII characters, %-encoding of octets not allowed in URIs, and Punycode-encoding of domain names.

### (profcim) StringIRI primitive

An IRI (Internationalized Resource Identifier) within an RDF graph is a Unicode string that conforms to the syntax defined in RFC 3987.

The primitive is serialized as literal without language support.

IRIs in the RDF abstract syntax must be absolute, and may contain a fragment identifier.

IRI equality: Two IRIs are equal if and only if they are equivalent under Simple String Comparison according to section 5.1 of [RFC3987]. Further normalization must not be performed when comparing IRIs for equality.

IRIs are a generalization of URIs [RFC3986] that permits a wider range of Unicode characters. Every absolute URI and URL is an IRI, but not every IRI is an URI. When IRIs are used in operations that are only defined for URIs, they must first be converted according to the mapping defined in section 3.1 of [RFC3987]. A notable example is retrieval over the HTTP protocol. The mapping involves UTF-8 encoding of non-ASCII characters, %-encoding of octets not allowed in URIs, and Punycode-encoding of domain names.

### (profcim) StringFixedLanguage primitive

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

The primitive is serialized as literal without language support.

### (profcim) URL primitive

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

### Package AdditionalClasses

#### General

#### (Description) PerLengthSequenceImpedance root class

Sequence impedance and admittance parameters per unit length, for transposed lines of 1, 2, or 3 phases. For 1-phase lines, define x=x0=xself. For 2-phase lines, define x=xs-xm and x0=xs+xm.

Table 346 shows all attributes of PerLengthSequenceImpedance.

Table 346 – Attributes of AdditionalClasses::PerLengthSequenceImpedance

| name | mult | type | description |
| --- | --- | --- | --- |
| b0ch | 0..1 | [SusceptancePerLength](#UML77) | Zero sequence shunt (charging) susceptance, per unit of length. |
| g0ch | 0..1 | [ConductancePerLength](#UML78) | Zero sequence shunt (charging) conductance, per unit of length. |
| r0 | 0..1 | [ResistancePerLength](#UML76) | Zero sequence series resistance, per unit of length. |
| x0 | 0..1 | [ReactancePerLength](#UML74) | Zero sequence series reactance, per unit of length. |

## Requirements and constraints

# Geographical location profile

## Package GeographicalLocationProfile

### General

This is the geographical location profile as in 61968-13.

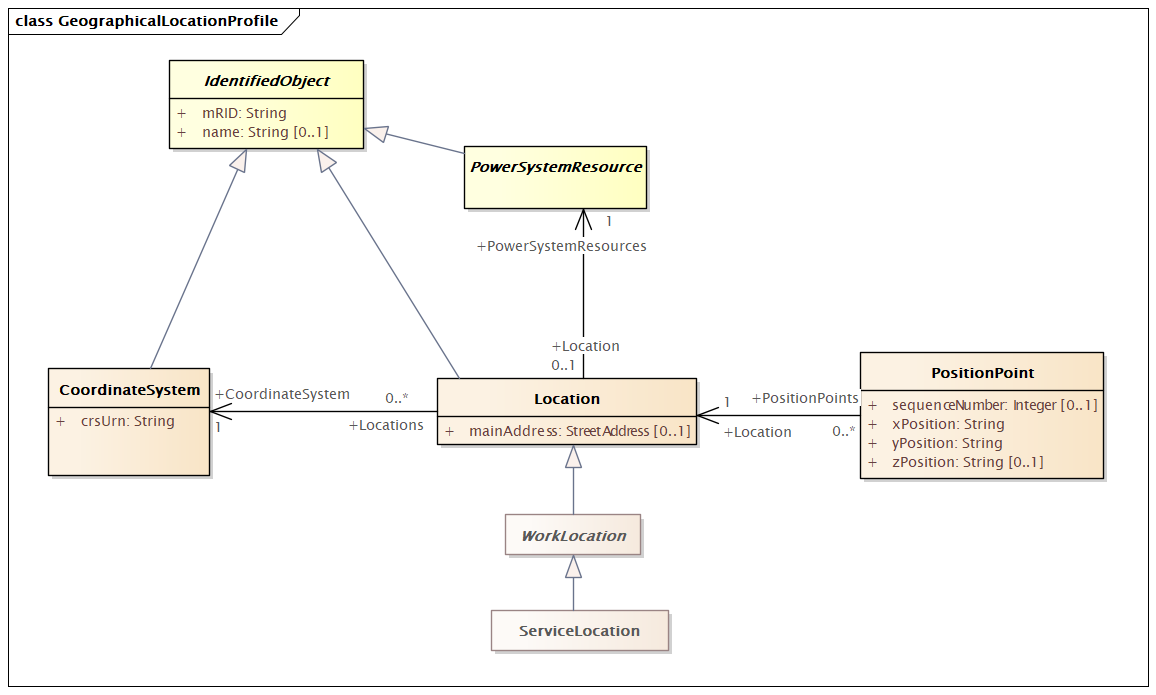


Figure 28 – Class diagram GeographicalLocationProfile::GeographicalLocationProfile

Figure 28: The diagram shows geographical location profile.

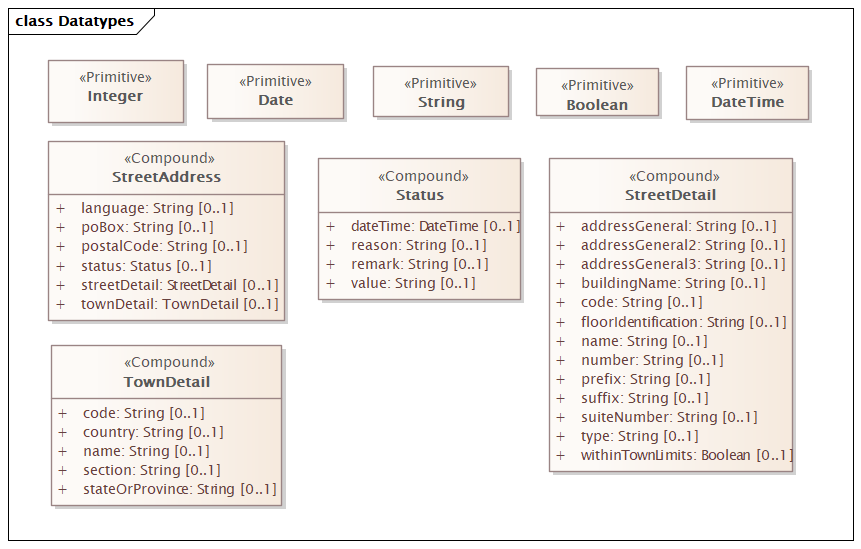


Figure 29 – Class diagram GeographicalLocationProfile::Datatypes

Figure 29: The diagram shows datatypes that are used by classes in the profile. Stereotypes are used to describe the datatypes. The following stereotypes are defined:

<<enumeration>> A list of permissible constant values.

<<Primitive>> The most basic data types used to compose all other data types.

<<CIMDatatype>> A datatype that contains a value attribute, an optional unit of measure and a unit multiplier. The unit and multiplier may be specified as a static variable initialized to the allowed value.

<<Compound>> A composite of Primitive, enumeration, CIMDatatype or other Compound classes, as long as the Compound classes do not recurse.

For all datatypes both positive and negative values are allowed unless stated otherwise for a particular datatype.

### (abstract) IdentifiedObject root class

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

Table 347 shows all attributes of IdentifiedObject.

Table 347 – Attributes of GeographicalLocationProfile::IdentifiedObject

| name | mult | type | description |
| --- | --- | --- | --- |
| mRID | 1..1 | [String](#UML883) | Master resource identifier issued by a model authority. The mRID is unique within an exchange context. Global uniqueness is easily achieved by using a UUID, as specified in RFC 4122, for the mRID. The use of UUID is strongly recommended.  For CIMXML data files in RDF syntax conforming to IEC 61970-552, the mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements. |
| name | 0..1 | [String](#UML883) | The name is any free human readable and possibly non unique text naming the object. |

### ServiceLocation

Inheritance path = [WorkLocation](#UML3125) : [Location](#UML3124) : [IdentifiedObject](#UML874)

A real estate location, commonly referred to as premises.

Table 348 shows all attributes of ServiceLocation.

Table 348 – Attributes of GeographicalLocationProfile::ServiceLocation

| name | mult | type | description |
| --- | --- | --- | --- |
| mainAddress | 0..1 | [StreetAddress](#UML876) | inherited from: [Location](#UML3124) |
| mRID | 1..1 | [String](#UML883) | inherited from: [IdentifiedObject](#UML874) |
| name | 0..1 | [String](#UML883) | inherited from: [IdentifiedObject](#UML874) |

Table 349 shows all association ends of ServiceLocation with other classes.

Table 349 – Association ends of GeographicalLocationProfile::ServiceLocation with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | CoordinateSystem | 1..1 | [CoordinateSystem](#UML3123) | inherited from: [Location](#UML3124) |
| 0..1 | PowerSystemResources | 1..1 | [PowerSystemResource](#UML3127) | inherited from: [Location](#UML3124) |

### (abstract) WorkLocation

Inheritance path = [Location](#UML3124) : [IdentifiedObject](#UML874)

Information about a particular location for various forms of work.

Table 350 shows all attributes of WorkLocation.

Table 350 – Attributes of GeographicalLocationProfile::WorkLocation

| name | mult | type | description |
| --- | --- | --- | --- |
| mainAddress | 0..1 | [StreetAddress](#UML876) | inherited from: [Location](#UML3124) |
| mRID | 1..1 | [String](#UML883) | inherited from: [IdentifiedObject](#UML874) |
| name | 0..1 | [String](#UML883) | inherited from: [IdentifiedObject](#UML874) |

Table 351 shows all association ends of WorkLocation with other classes.

Table 351 – Association ends of GeographicalLocationProfile::WorkLocation with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | CoordinateSystem | 1..1 | [CoordinateSystem](#UML3123) | inherited from: [Location](#UML3124) |
| 0..1 | PowerSystemResources | 1..1 | [PowerSystemResource](#UML3127) | inherited from: [Location](#UML3124) |

### CoordinateSystem

Inheritance path = [IdentifiedObject](#UML874)

Coordinate reference system.

Table 352 shows all attributes of CoordinateSystem.

Table 352 – Attributes of GeographicalLocationProfile::CoordinateSystem

| name | mult | type | description |
| --- | --- | --- | --- |
| crsUrn | 1..1 | [String](#UML883) | A Uniform Resource Name (URN) for the coordinate reference system (crs) used to define 'Location.PositionPoints'.  An example would be the European Petroleum Survey Group (EPSG) code for a coordinate reference system, defined in URN under the Open Geospatial Consortium (OGC) namespace as: urn:ogc:def:crs:EPSG::XXXX, where XXXX is an EPSG code (a full list of codes can be found at the EPSG Registry web site http://www.epsg-registry.org/). To define the coordinate system as being WGS84 (latitude, longitude) using an EPSG OGC, this attribute would be urn:ogc:def:crs:EPSG::4236.  A profile should limit this code to a set of allowed URNs agreed to by all sending and receiving parties. |
| mRID | 1..1 | [String](#UML883) | inherited from: [IdentifiedObject](#UML874) |
| name | 0..1 | [String](#UML883) | inherited from: [IdentifiedObject](#UML874) |

### Location

Inheritance path = [IdentifiedObject](#UML874)

The place, scene, or point of something where someone or something has been, is, and/or will be at a given moment in time. It can be defined with one or more position points (coordinates) in a given coordinate system.

Table 353 shows all attributes of Location.

Table 353 – Attributes of GeographicalLocationProfile::Location

| name | mult | type | description |
| --- | --- | --- | --- |
| mainAddress | 0..1 | [StreetAddress](#UML876) | Main address of the location. |
| mRID | 1..1 | [String](#UML883) | inherited from: [IdentifiedObject](#UML874) |
| name | 0..1 | [String](#UML883) | inherited from: [IdentifiedObject](#UML874) |

Table 354 shows all association ends of Location with other classes.

Table 354 – Association ends of GeographicalLocationProfile::Location with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | CoordinateSystem | 1..1 | [CoordinateSystem](#UML3123) | Coordinate system used to describe position points of this location. |
| 0..1 | PowerSystemResources | 1..1 | [PowerSystemResource](#UML3127) | All power system resources at this location. |

### PositionPoint root class

Set of spatial coordinates that determine a point, defined in the coordinate system specified in 'Location.CoordinateSystem'. Use a single position point instance to describe a point-oriented location. Use a sequence of position points to describe a line-oriented object (physical location of non-point oriented objects like cables or lines), or area of an object (like a substation or a geographical zone - in this case, have first and last position point with the same values).

Table 355 shows all attributes of PositionPoint.

Table 355 – Attributes of GeographicalLocationProfile::PositionPoint

| name | mult | type | description |
| --- | --- | --- | --- |
| sequenceNumber | 0..1 | [Integer](#UML882) | Zero-relative sequence number of this point within a series of points. |
| xPosition | 1..1 | [String](#UML883) | X axis position. |
| yPosition | 1..1 | [String](#UML883) | Y axis position. |
| zPosition | 0..1 | [String](#UML883) | (if applicable) Z axis position. |

Table 356 shows all association ends of PositionPoint with other classes.

Table 356 – Association ends of GeographicalLocationProfile::PositionPoint with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Location | 1..1 | [Location](#UML3124) | Location described by this position point. |

### (abstract) PowerSystemResource

Inheritance path = [IdentifiedObject](#UML874)

A power system resource (PSR) 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.

Table 357 shows all attributes of PowerSystemResource.

Table 357 – Attributes of GeographicalLocationProfile::PowerSystemResource

| name | mult | type | description |
| --- | --- | --- | --- |
| mRID | 1..1 | [String](#UML883) | inherited from: [IdentifiedObject](#UML874) |
| name | 0..1 | [String](#UML883) | inherited from: [IdentifiedObject](#UML874) |

### StreetAddress compound

General purpose street and postal address information.

Table 358 shows all attributes of StreetAddress.

Table 358 – Attributes of GeographicalLocationProfile::StreetAddress

| name | mult | type | description |
| --- | --- | --- | --- |
| streetDetail | 0..1 | [StreetDetail](#UML879) | Street detail. |
| townDetail | 0..1 | [TownDetail](#UML877) | Town detail. |
| status | 0..1 | [Status](#UML878) | Status of this address. |
| postalCode | 0..1 | [String](#UML883) | Postal code for the address. |
| poBox | 0..1 | [String](#UML883) | Post office box. |
| language | 0..1 | [String](#UML883) | The language in which the address is specified, using ISO 639-1 two digit language code. |

### TownDetail compound

Town details, in the context of address.

Table 359 shows all attributes of TownDetail.

Table 359 – Attributes of GeographicalLocationProfile::TownDetail

| name | mult | type | description |
| --- | --- | --- | --- |
| code | 0..1 | [String](#UML883) | Town code. |
| section | 0..1 | [String](#UML883) | Town section. For example, it is common for there to be 36 sections per township. |
| name | 0..1 | [String](#UML883) | Town name. |
| stateOrProvince | 0..1 | [String](#UML883) | Name of the state or province. |
| country | 0..1 | [String](#UML883) | Name of the country. |

### Status compound

Current status information relevant to an entity.

Table 360 shows all attributes of Status.

Table 360 – Attributes of GeographicalLocationProfile::Status

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [String](#UML883) | Status value at 'dateTime'; prior status changes may have been kept in instances of activity records associated with the object to which this status applies. |
| dateTime | 0..1 | [DateTime](#UML880) | Date and time for which status 'value' applies. |
| remark | 0..1 | [String](#UML883) | Pertinent information regarding the current 'value', as free form text. |
| reason | 0..1 | [String](#UML883) | Reason code or explanation for why an object went to the current status 'value'. |

### StreetDetail compound

Street details, in the context of address.

Table 361 shows all attributes of StreetDetail.

Table 361 – Attributes of GeographicalLocationProfile::StreetDetail

| name | mult | type | description |
| --- | --- | --- | --- |
| number | 0..1 | [String](#UML883) | Designator of the specific location on the street. |
| name | 0..1 | [String](#UML883) | Name of the street. |
| suffix | 0..1 | [String](#UML883) | Suffix to the street name. For example: North, South, East, West. |
| prefix | 0..1 | [String](#UML883) | Prefix to the street name. For example: North, South, East, West. |
| type | 0..1 | [String](#UML883) | Type of street. Examples include: street, circle, boulevard, avenue, road, drive, etc. |
| code | 0..1 | [String](#UML883) | (if applicable) Utilities often make use of external reference systems, such as those of the town-planner's department or surveyor general's mapping system, that allocate global reference codes to streets. |
| buildingName | 0..1 | [String](#UML883) | (if applicable) In certain cases the physical location of the place of interest does not have a direct point of entry from the street, but may be located inside a larger structure such as a building, complex, office block, apartment, etc. |
| suiteNumber | 0..1 | [String](#UML883) | Number of the apartment or suite. |
| addressGeneral | 0..1 | [String](#UML883) | First line of a free form address or some additional address information (for example a mail stop). |
| addressGeneral2 | 0..1 | [String](#UML883) | (if applicable) Second line of a free form address. |
| addressGeneral3 | 0..1 | [String](#UML883) | (if applicable) Third line of a free form address. |
| withinTownLimits | 0..1 | [Boolean](#UML881) | True if this street is within the legal geographical boundaries of the specified town (default). |
| floorIdentification | 0..1 | [String](#UML883) | The identification by name or number, expressed as text, of the floor in the building as part of this address. |

### DateTime primitive

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

### Boolean primitive

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

### Integer primitive

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

### String primitive

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

### Date primitive

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

# Diagram Layout profile

## Package DiagramLayoutProfile

### General

This standard specifies an exchange profile for schematics with the following characteristics.

A generic method for linking the diagram object to the domain data. Domain data and diagram layout data can be exchanged together with the domain data, or separately from each other with the assumption that domain data will have already been imported should diagram layout data be imported separately.

It does not require or imply a specific domain data format. Therefore, it supports domain data modelled according to the IEC 61970-301 Common information model (CIM) that is exchanged in the IEC 61970-501 format (CIM RDF schema). Diagram layout data references domain data in compliance with IEC/TR 62541-1, OPC Unified Architecture).

The intended usage of this standard is that the source of a diagram will use the standard to encode the layout of their diagram as they created it. It is then incumbent upon any receiver to supply the means of using that diagram layout in their system.

- In the simplest situation, if the source uses Switch object placements, the receiver creates a generic Switch template in the receiver's system that will be used to render Switches (i.e. there is no exchange of Switch templates).

- In more complex situations, a 1:1 correspondence may not be desirable, and receivers may have to create generic transformations in order to use exchanges.

- However, in both situations, it is expected that once an investment is made in a strategy for rendering, repeated imports may be carried out automatically.

Supports the exchange of diagram objects that have no relationship to domain data, i.e., pure static background objects.

Supports multiple representations of the same domain object in the same or different diagrams.

Allows the diagram to be used as the unit of exchange, providing a straightforward approach to partial exchange, or an exchange between systems that have a separate model and repository for diagram layout data.

Supports assignment of diagram objects to layers or other means, for showing or hiding information based on zoom level and/or user interest.

A generic method for proprietary extensions to enable full round tripping (export and import back into the same system) without information loss within a system, and without breaking the standard exchange format.

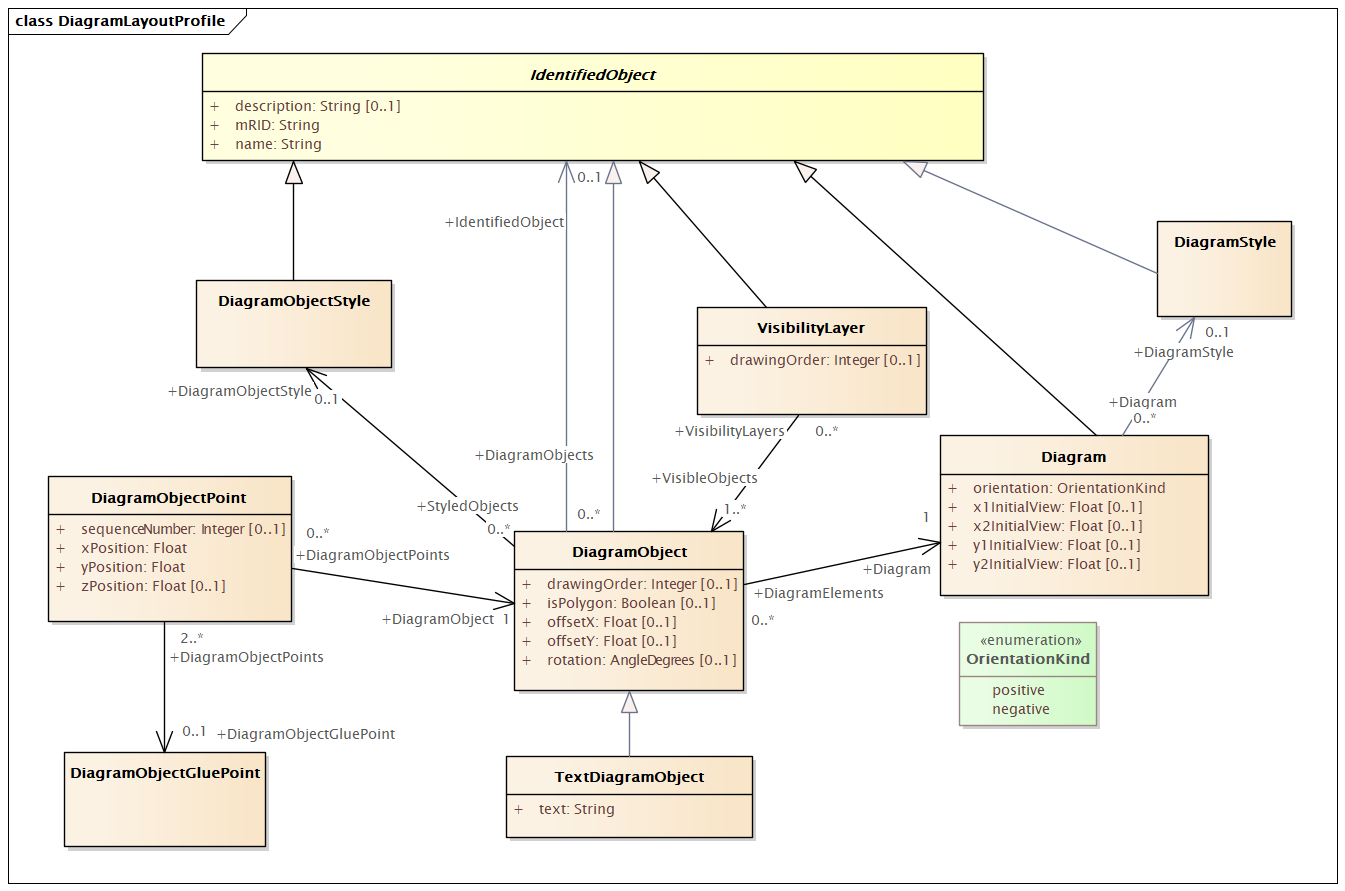


Figure 30 – Class diagram DiagramLayoutProfile::DiagramLayoutProfile

Figure 30: The diagram shows Diagram Layout Profile.

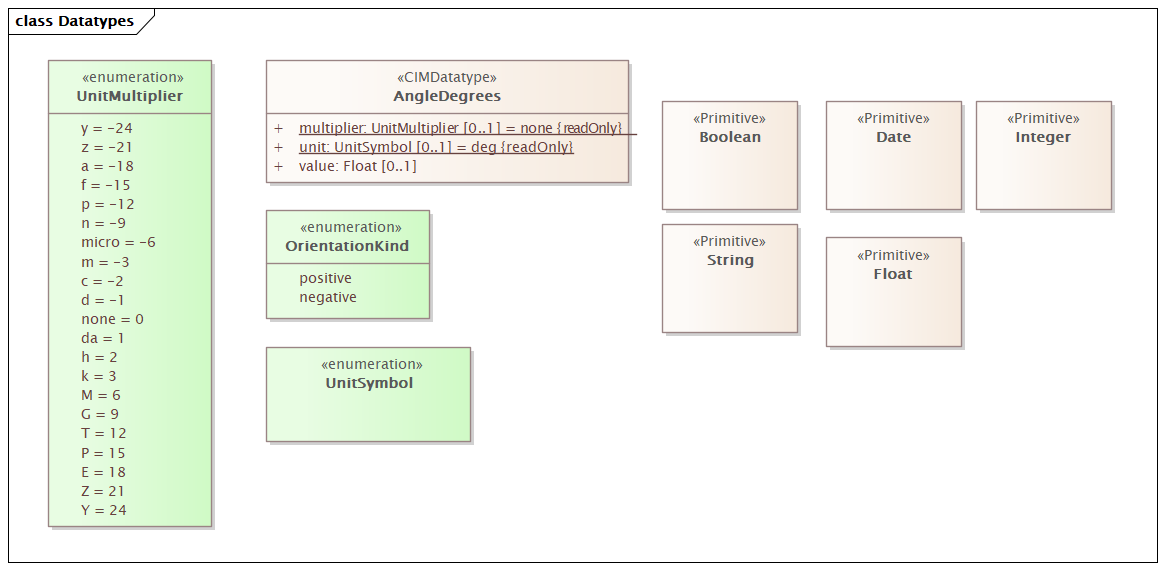


Figure 31 – Class diagram DiagramLayoutProfile::Datatypes

Figure 31: The diagram shows datatypes that are used by classes in the profile. Stereotypes are used to describe the datatypes. The following stereotypes are defined:

<<enumeration>> A list of permissible constant values.

<<Primitive>> The most basic data types used to compose all other data types.

<<CIMDatatype>> A datatype that contains a value attribute, an optional unit of measure and a unit multiplier. The unit and multiplier may be specified as a static variable initialized to the allowed value.

<<Compound>> A composite of Primitive, enumeration, CIMDatatype or other Compound classes, as long as the Compound classes do not recurse.

For all datatypes both positive and negative values are allowed unless stated otherwise for a particular datatype.

### Diagram

Inheritance path = [IdentifiedObject](#UML608)

The diagram being exchanged. The coordinate system is a standard Cartesian coordinate system and the orientation attribute defines the orientation. The initial view related attributes can be used to specify an initial view with the x,y coordinates of the diagonal points.

Table 362 shows all attributes of Diagram.

Table 362 – Attributes of DiagramLayoutProfile::Diagram

| name | mult | type | description |
| --- | --- | --- | --- |
| orientation | 1..1 | [OrientationKind](#UML609) | Coordinate system orientation of the diagram. A positive orientation gives standard “right-hand” orientation, with negative orientation indicating a “left-hand” orientation. For 2D diagrams, a positive orientation will result in X values increasing from left to right and Y values increasing from bottom to top. A negative orientation gives the “left-hand” orientation (favoured by computer graphics displays) with X values increasing from left to right and Y values increasing from top to bottom. |
| x1InitialView | 0..1 | [Float](#UML615) | X coordinate of the first corner of the initial view. |
| x2InitialView | 0..1 | [Float](#UML615) | X coordinate of the second corner of the initial view. |
| y1InitialView | 0..1 | [Float](#UML615) | Y coordinate of the first corner of the initial view. |
| y2InitialView | 0..1 | [Float](#UML615) | Y coordinate of the second corner of the initial view. |
| mRID | 1..1 | [String](#UML617) | inherited from: [IdentifiedObject](#UML608) |
| name | 1..1 | [String](#UML617) | inherited from: [IdentifiedObject](#UML608) |
| description | 0..1 | [String](#UML617) | inherited from: [IdentifiedObject](#UML608) |

Table 363 shows all association ends of Diagram with other classes.

Table 363 – Association ends of DiagramLayoutProfile::Diagram with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | DiagramStyle | 0..1 | [DiagramStyle](#UML2564) | A Diagram may have a DiagramStyle. |

### DiagramObject

Inheritance path = [IdentifiedObject](#UML608)

An object that defines one or more points in a given space. This object can be associated with anything that specializes IdentifiedObject. For single line diagrams such objects typically include such items as analog values, breakers, disconnectors, power transformers, and transmission lines.

Table 364 shows all attributes of DiagramObject.

Table 364 – Attributes of DiagramLayoutProfile::DiagramObject

| name | mult | type | description |
| --- | --- | --- | --- |
| drawingOrder | 0..1 | [Integer](#UML616) | The drawing order of this element. The higher the number, the later the element is drawn in sequence. This is used to ensure that elements that overlap are rendered in the correct order. |
| isPolygon | 0..1 | [Boolean](#UML613) | Defines whether or not the diagram objects points define the boundaries of a polygon or the routing of a polyline. If this value is true then a receiving application should consider the first and last points to be connected. |
| offsetX | 0..1 | [Float](#UML615) | The offset in the X direction. This is used for defining the offset from centre for rendering an icon (the default is that a single point specifies the centre of the icon).  The offset is in per-unit with 0 indicating there is no offset from the horizontal centre of the icon. -0.5 indicates it is offset by 50% to the left and 0.5 indicates an offset of 50% to the right. |
| offsetY | 0..1 | [Float](#UML615) | The offset in the Y direction. This is used for defining the offset from centre for rendering an icon (the default is that a single point specifies the centre of the icon).  The offset is in per-unit with 0 indicating there is no offset from the vertical centre of the icon. The offset direction is dependent on the orientation of the diagram, with -0.5 and 0.5 indicating an offset of +/- 50% on the vertical axis. |
| rotation | 0..1 | [AngleDegrees](#UML612) | Sets the angle of rotation of the diagram object. Zero degrees is pointing to the top of the diagram. Rotation is clockwise. DiagramObject.rotation=0 has the following meaning: The connection point of an element which has one terminal is pointing to the top side of the diagram. The connection point "From side" of an element which has more than one terminal is pointing to the top side of the diagram.  DiagramObject.rotation=90 has the following meaning: The connection point of an element which has one terminal is pointing to the right hand side of the diagram. The connection point "From side" of an element which has more than one terminal is pointing to the right hand side of the diagram. |
| mRID | 1..1 | [String](#UML617) | inherited from: [IdentifiedObject](#UML608) |
| name | 1..1 | [String](#UML617) | inherited from: [IdentifiedObject](#UML608) |
| description | 0..1 | [String](#UML617) | inherited from: [IdentifiedObject](#UML608) |

Table 365 shows all association ends of DiagramObject with other classes.

Table 365 – Association ends of DiagramLayoutProfile::DiagramObject with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Diagram | 1..1 | [Diagram](#UML2568) | A diagram object is part of a diagram. |
| 0..\* | DiagramObjectStyle | 0..1 | [DiagramObjectStyle](#UML2565) | A diagram object has a style associated that provides a reference for the style used in the originating system. |
| 0..\* | IdentifiedObject | 0..1 | [IdentifiedObject](#UML608) | The domain object to which this diagram object is associated. |

### DiagramObjectGluePoint root class

This is used for grouping diagram object points from different diagram objects that are considered to be glued together in a diagram even if they are not at the exact same coordinates.

### DiagramObjectPoint root class

A point in a given space defined by 3 coordinates and associated to a diagram object. The coordinates may be positive or negative as the origin does not have to be in the corner of a diagram.

Table 366 shows all attributes of DiagramObjectPoint.

Table 366 – Attributes of DiagramLayoutProfile::DiagramObjectPoint

| name | mult | type | description |
| --- | --- | --- | --- |
| sequenceNumber | 0..1 | [Integer](#UML616) | The sequence position of the point, used for defining the order of points for diagram objects acting as a polyline or polygon with more than one point. The attribute shall be a positive value. |
| xPosition | 1..1 | [Float](#UML615) | The X coordinate of this point. |
| yPosition | 1..1 | [Float](#UML615) | The Y coordinate of this point. |
| zPosition | 0..1 | [Float](#UML615) | The Z coordinate of this point. |

Table 367 shows all association ends of DiagramObjectPoint with other classes.

Table 367 – Association ends of DiagramLayoutProfile::DiagramObjectPoint with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | DiagramObject | 1..1 | [DiagramObject](#UML2566) | The diagram object with which the points are associated. |
| 2..\* | DiagramObjectGluePoint | 0..1 | [DiagramObjectGluePoint](#UML606) | The 'glue' point to which this point is associated. |

### DiagramObjectStyle

Inheritance path = [IdentifiedObject](#UML608)

A reference to a style used by the originating system for a diagram object. A diagram object style describes information such as line thickness, shape such as circle or rectangle etc, and colour.

Table 368 shows all attributes of DiagramObjectStyle.

Table 368 – Attributes of DiagramLayoutProfile::DiagramObjectStyle

| name | mult | type | description |
| --- | --- | --- | --- |
| mRID | 1..1 | [String](#UML617) | inherited from: [IdentifiedObject](#UML608) |
| name | 1..1 | [String](#UML617) | inherited from: [IdentifiedObject](#UML608) |
| description | 0..1 | [String](#UML617) | inherited from: [IdentifiedObject](#UML608) |

### DiagramStyle

Inheritance path = [IdentifiedObject](#UML608)

The diagram style refers to a style used by the originating system for a diagram. A diagram style describes information such as schematic, geographic, etc.

Table 369 shows all attributes of DiagramStyle.

Table 369 – Attributes of DiagramLayoutProfile::DiagramStyle

| name | mult | type | description |
| --- | --- | --- | --- |
| mRID | 1..1 | [String](#UML617) | inherited from: [IdentifiedObject](#UML608) |
| name | 1..1 | [String](#UML617) | inherited from: [IdentifiedObject](#UML608) |
| description | 0..1 | [String](#UML617) | inherited from: [IdentifiedObject](#UML608) |

### (abstract) IdentifiedObject root class

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

Table 370 shows all attributes of IdentifiedObject.

Table 370 – Attributes of DiagramLayoutProfile::IdentifiedObject

| name | mult | type | description |
| --- | --- | --- | --- |
| mRID | 1..1 | [String](#UML617) | Master resource identifier issued by a model authority. The mRID is unique within an exchange context. Global uniqueness is easily achieved by using a UUID, as specified in RFC 4122, for the mRID. The use of UUID is strongly recommended.  For CIMXML data files in RDF syntax conforming to IEC 61970-552, the mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements. |
| name | 1..1 | [String](#UML617) | The name is any free human readable and possibly non unique text naming the object. |
| description | 0..1 | [String](#UML617) | 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. |

### TextDiagramObject

Inheritance path = [DiagramObject](#UML2566) : [IdentifiedObject](#UML608)

A diagram object for placing free-text or text derived from an associated domain object.

Table 371 shows all attributes of TextDiagramObject.

Table 371 – Attributes of DiagramLayoutProfile::TextDiagramObject

| name | mult | type | description |
| --- | --- | --- | --- |
| text | 1..1 | [String](#UML617) | The text that is displayed by this text diagram object. |
| drawingOrder | 0..1 | [Integer](#UML616) | inherited from: [DiagramObject](#UML2566) |
| isPolygon | 0..1 | [Boolean](#UML613) | inherited from: [DiagramObject](#UML2566) |
| offsetX | 0..1 | [Float](#UML615) | inherited from: [DiagramObject](#UML2566) |
| offsetY | 0..1 | [Float](#UML615) | inherited from: [DiagramObject](#UML2566) |
| rotation | 0..1 | [AngleDegrees](#UML612) | inherited from: [DiagramObject](#UML2566) |
| mRID | 1..1 | [String](#UML617) | inherited from: [IdentifiedObject](#UML608) |
| name | 1..1 | [String](#UML617) | inherited from: [IdentifiedObject](#UML608) |
| description | 0..1 | [String](#UML617) | inherited from: [IdentifiedObject](#UML608) |

Table 372 shows all association ends of TextDiagramObject with other classes.

Table 372 – Association ends of DiagramLayoutProfile::TextDiagramObject with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | Diagram | 1..1 | [Diagram](#UML2568) | inherited from: [DiagramObject](#UML2566) |
| 0..\* | DiagramObjectStyle | 0..1 | [DiagramObjectStyle](#UML2565) | inherited from: [DiagramObject](#UML2566) |
| 0..\* | IdentifiedObject | 0..1 | [IdentifiedObject](#UML608) | inherited from: [DiagramObject](#UML2566) |

### VisibilityLayer

Inheritance path = [IdentifiedObject](#UML608)

Layers are typically used for grouping diagram objects according to themes and scales. Themes are used to display or hide certain information (e.g., lakes, borders), while scales are used for hiding or displaying information depending on the current zoom level (hide text when it is too small to be read, or when it exceeds the screen size). This is also called de-cluttering.

CIM based graphics exchange supports an m:n relationship between diagram objects and layers. The importing system shall convert an m:n case into an appropriate 1:n representation if the importing system does not support m:n.

Table 373 shows all attributes of VisibilityLayer.

Table 373 – Attributes of DiagramLayoutProfile::VisibilityLayer

| name | mult | type | description |
| --- | --- | --- | --- |
| drawingOrder | 0..1 | [Integer](#UML616) | The drawing order for this layer. The higher the number, the later the layer and the objects within it are rendered. |
| mRID | 1..1 | [String](#UML617) | inherited from: [IdentifiedObject](#UML608) |
| name | 1..1 | [String](#UML617) | inherited from: [IdentifiedObject](#UML608) |
| description | 0..1 | [String](#UML617) | inherited from: [IdentifiedObject](#UML608) |

Table 374 shows all association ends of VisibilityLayer with other classes.

Table 374 – Association ends of DiagramLayoutProfile::VisibilityLayer with other classes

| mult from | name | mult to | type | description |
| --- | --- | --- | --- | --- |
| 0..\* | VisibleObjects | 1..\* | [DiagramObject](#UML2566) | A visibility layer can contain one or more diagram objects. |

### OrientationKind enumeration

The orientation of the coordinate system with respect to top, left, and the coordinate number system.

Table 375 shows all literals of OrientationKind.

Table 375 – Literals of DiagramLayoutProfile::OrientationKind

| literal | value | description |
| --- | --- | --- |
| positive |  | For 2D diagrams, a positive orientation will result in X values increasing from left to right and Y values increasing from bottom to top. This is also known as a right hand orientation. |
| negative |  | For 2D diagrams, a negative orientation gives the left-hand orientation (favoured by computer graphics displays) with X values increasing from left to right and Y values increasing from top to bottom. This is also known as a left hand orientation. |

### UnitMultiplier enumeration

The unit multipliers defined for the CIM. When applied to unit symbols, the unit symbol is treated as a derived unit. Regardless of the contents of the unit symbol text, the unit symbol shall be treated as if it were a single-character unit symbol. Unit symbols should not contain multipliers, and it should be left to the multiplier to define the multiple for an entire data type.

For example, if a unit symbol is "m2Pers" and the multiplier is "k", then the value is k(m\*\*2/s), and the multiplier applies to the entire final value, not to any individual part of the value. This can be conceptualized by substituting a derived unit symbol for the unit type. If one imagines that the symbol "Þ" represents the derived unit "m2Pers", then applying the multiplier "k" can be conceptualized simply as "kÞ".

For example, the SI unit for mass is "kg" and not "g". If the unit symbol is defined as "kg", then the multiplier is applied to "kg" as a whole and does not replace the "k" in front of the "g". In this case, the multiplier of "m" would be used with the unit symbol of "kg" to represent one gram. As a text string, this violates the instructions in IEC 80000-1. However, because the unit symbol in CIM is treated as a derived unit instead of as an SI unit, it makes more sense to conceptualize the "kg" as if it were replaced by one of the proposed replacements for the SI mass symbol. If one imagines that the "kg" were replaced by a symbol "Þ", then it is easier to conceptualize the multiplier "m" as creating the proper unit "mÞ", and not the forbidden unit "mkg".

Table 376 shows all literals of UnitMultiplier.

Table 376 – Literals of DiagramLayoutProfile::UnitMultiplier

| literal | value | description |
| --- | --- | --- |
| y | -24 | Yocto 10\*\*-24. |
| z | -21 | Zepto 10\*\*-21. |
| a | -18 | Atto 10\*\*-18. |
| f | -15 | Femto 10\*\*-15. |
| p | -12 | Pico 10\*\*-12. |
| n | -9 | Nano 10\*\*-9. |
| micro | -6 | Micro 10\*\*-6. |
| m | -3 | Milli 10\*\*-3. |
| c | -2 | Centi 10\*\*-2. |
| d | -1 | Deci 10\*\*-1. |
| none | 0 | No multiplier or equivalently multiply by 1. |
| da | 1 | Deca 10\*\*1. |
| h | 2 | Hecto 10\*\*2. |
| k | 3 | Kilo 10\*\*3. |
| M | 6 | Mega 10\*\*6. |
| G | 9 | Giga 10\*\*9. |
| T | 12 | Tera 10\*\*12. |
| P | 15 | Peta 10\*\*15. |
| E | 18 | Exa 10\*\*18. |
| Z | 21 | Zetta 10\*\*21. |
| Y | 24 | Yotta 10\*\*24. |

### UnitSymbol enumeration

The derived units defined for usage in the CIM. In some cases, the derived unit is equal to an SI unit. Whenever possible, the standard derived symbol is used instead of the formula for the derived unit. For example, the unit symbol Farad is defined as "F" instead of "CPerV". In cases where a standard symbol does not exist for a derived unit, the formula for the unit is used as the unit symbol. For example, density does not have a standard symbol and so it is represented as "kgPerm3". With the exception of the "kg", which is an SI unit, the unit symbols do not contain multipliers and therefore represent the base derived unit to which a multiplier can be applied as a whole.

Every unit symbol is treated as an unparseable text as if it were a single-letter symbol. The meaning of each unit symbol is defined by the accompanying descriptive text and not by the text contents of the unit symbol.

To allow the widest possible range of serializations without requiring special character handling, several substitutions are made which deviate from the format described in IEC 80000-1. The division symbol "/" is replaced by the letters "Per". Exponents are written in plain text after the unit as "m3" instead of being formatted as "m" with a superscript of 3 or introducing a symbol as in "m^3". The degree symbol "°" is replaced with the letters "deg". Any clarification of the meaning for a substitution is included in the description for the unit symbol.

Non-SI units are included in list of unit symbols to allow sources of data to be correctly labelled with their non-SI units (for example, a GPS sensor that is reporting numbers that represent feet instead of meters). This allows software to use the unit symbol information correctly convert and scale the raw data of those sources into SI-based units.

The integer values are used for harmonization with IEC 61850.

Table 377 shows all literals of UnitSymbol.

Table 377 – Literals of DiagramLayoutProfile::UnitSymbol

| literal | value | description |
| --- | --- | --- |
| none | 0 | Dimension less quantity, e.g. count, per unit, etc. |
| m | 2 | Length in metres. |
| kg | 3 | Mass in kilograms. Note: multiplier “k” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| s | 4 | Time in seconds. |
| A | 5 | Current in amperes. |
| K | 6 | Temperature in kelvins. |
| mol | 7 | Amount of substance in moles. |
| cd | 8 | Luminous intensity in candelas. |
| deg | 9 | Plane angle in degrees. |
| rad | 10 | Plane angle in radians (m/m). |
| sr | 11 | Solid angle in steradians (m2/m2). |
| Gy | 21 | Absorbed dose in grays (J/kg). |
| Bq | 22 | Radioactivity in becquerels (1/s). |
| degC | 23 | 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 from coulomb the symbol used in the UML is degC. The reason for not using °C is that the special character ° is difficult to manage in software. |
| Sv | 24 | Dose equivalent in sieverts (J/kg). |
| F | 25 | Electric capacitance in farads (C/V). |
| C | 26 | Electric charge in coulombs (A·s). |
| S | 27 | Conductance in siemens. |
| H | 28 | Electric inductance in henrys (Wb/A). |
| V | 29 | Electric potential in volts (W/A). |
| ohm | 30 | Electric resistance in ohms (V/A). |
| J | 31 | Energy in joules (N·m = C·V = W·s). |
| N | 32 | Force in newtons (kg·m/s²). |
| Hz | 33 | Frequency in hertz (1/s). |
| lx | 34 | Illuminance in lux (lm/m²). |
| lm | 35 | Luminous flux in lumens (cd·sr). |
| Wb | 36 | Magnetic flux in webers (V·s). |
| T | 37 | Magnetic flux density in teslas (Wb/m2). |
| W | 38 | Real power in watts (J/s). Electrical power may have real and reactive components. The real portion of electrical power (I²R or VIcos(phi)), is expressed in Watts. See also apparent power and reactive power. |
| Pa | 39 | Pressure in pascals (N/m²). Note: the absolute or relative measurement of pressure is implied with this entry. See below for more explicit forms. |
| m2 | 41 | Area in square metres (m²). |
| m3 | 42 | Volume in cubic metres (m³). |
| mPers | 43 | Velocity in metres per second (m/s). |
| mPers2 | 44 | Acceleration in metres per second squared (m/s²). |
| m3Pers | 45 | Volumetric flow rate in cubic metres per second (m³/s). |
| mPerm3 | 46 | Fuel efficiency in metres per cubic metres (m/m³). |
| kgm | 47 | Moment of mass in kilogram metres (kg·m) (first moment of mass). Note: multiplier “k” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| kgPerm3 | 48 | Density in kilogram/cubic metres (kg/m³). Note: multiplier “k” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| m2Pers | 49 | Viscosity in square metres / second (m²/s). |
| WPermK | 50 | Thermal conductivity in watt/metres kelvin. |
| JPerK | 51 | Heat capacity in joules/kelvin. |
| ppm | 52 | Concentration in parts per million. |
| rotPers | 53 | Rotations per second (1/s). See also Hz (1/s). |
| radPers | 54 | Angular velocity in radians per second (rad/s). |
| WPerm2 | 55 | Heat flux density, irradiance, watts per square metre. |
| JPerm2 | 56 | Insulation energy density, joules per square metre or watt second per square metre. |
| SPerm | 57 | Conductance per length (F/m). |
| KPers | 58 | Temperature change rate in kelvins per second. |
| PaPers | 59 | Pressure change rate in pascals per second. |
| JPerkgK | 60 | Specific heat capacity, specific entropy, joules per kilogram Kelvin. |
| VA | 61 | Apparent power in volt amperes. See also real power and reactive power. |
| VAr | 63 | Reactive power in volt amperes reactive. The “reactive” or “imaginary” component of electrical power (VIsin(phi)). (See also real power and apparent power).  Note: Different meter designs use different methods to arrive at their results. Some meters may compute reactive power as an arithmetic value, while others compute the value vectorially. The data consumer should determine the method in use and the suitability of the measurement for the intended purpose. |
| cosPhi | 65 | Power factor, dimensionless.  Note 1: This definition of power factor only holds for balanced systems. See the alternative definition under code 153.  Note 2 : Beware of differing sign conventions in use between the IEC and EEI. It is assumed that the data consumer understands the type of meter in use and the sign convention in use by the utility. |
| Vs | 66 | Volt seconds (Ws/A). |
| V2 | 67 | Volt squared (W²/A²). |
| As | 68 | Ampere seconds (A·s). |
| A2 | 69 | Amperes squared (A²). |
| A2s | 70 | Ampere squared time in square amperes (A²s). |
| VAh | 71 | Apparent energy in volt ampere hours. |
| Wh | 72 | Real energy in watt hours. |
| VArh | 73 | Reactive energy in volt ampere reactive hours. |
| VPerHz | 74 | Magnetic flux in volt per hertz. |
| HzPers | 75 | Rate of change of frequency in hertz per second. |
| character | 76 | Number of characters. |
| charPers | 77 | Data rate (baud) in characters per second. |
| kgm2 | 78 | Moment of mass in kilogram square metres (kg·m²) (Second moment of mass, commonly called the moment of inertia). Note: multiplier “k” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| dB | 79 | Sound pressure level in decibels. Note: multiplier “d” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| WPers | 81 | Ramp rate in watts per second. |
| lPers | 82 | Volumetric flow rate in litres per second. |
| dBm | 83 | Power level (logarithmic ratio of signal strength , Bel-mW), normalized to 1mW. Note: multiplier “d” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| h | 84 | Time in hours, hour = 60 min = 3600 s. |
| min | 85 | Time in minutes, minute = 60 s. |
| Q | 100 | Quantity power, Q. |
| Qh | 101 | Quantity energy, Qh. |
| ohmm | 102 | Resistivity, ohm metres, (rho). |
| APerm | 103 | A/m, magnetic field strength, amperes per metre. |
| V2h | 104 | Volt-squared hour, volt-squared-hours. |
| A2h | 105 | Ampere-squared hour, ampere-squared hour. |
| Ah | 106 | Ampere-hours, ampere-hours. |
| count | 111 | Amount of substance, Counter value. |
| ft3 | 119 | Volume, cubic feet. |
| m3Perh | 125 | Volumetric flow rate, cubic metres per hour. |
| gal | 128 | Volume in gallons, US gallon (1 gal = 231 in3 = 128 fl ounce). |
| Btu | 132 | Energy, British Thermal Units. |
| l | 134 | Volume in litres, litre = dm3 = m3/1000. |
| lPerh | 137 | Volumetric flow rate, litres per hour. |
| lPerl | 143 | Concentration, The ratio of the volume of a solute divided by the volume of the solution. Note: Users may need use a prefix such a ‘µ’ to express a quantity such as ‘µL/L’. |
| gPerg | 144 | Concentration, The ratio of the mass of a solute divided by the mass of the solution. Note: Users may need use a prefix such a ‘µ’ to express a quantity such as ‘µg/g’. |
| molPerm3 | 145 | Concentration, The amount of substance concentration, (c), the amount of solvent in moles divided by the volume of solution in m³. |
| molPermol | 146 | Concentration, Molar fraction, the ratio of the molar amount of a solute divided by the molar amount of the solution. |
| molPerkg | 147 | Concentration, Molality, the amount of solute in moles and the amount of solvent in kilograms. |
| sPers | 149 | Time, Ratio of time. Note: Users may need to supply a prefix such as ‘µ’ to show rates such as ‘µs/s’. |
| HzPerHz | 150 | Frequency, rate of frequency change. Note: Users may need to supply a prefix such as ‘m’ to show rates such as ‘mHz/Hz’. |
| VPerV | 151 | Voltage, ratio of voltages. Note: Users may need to supply a prefix such as ‘m’ to show rates such as ‘mV/V’. |
| APerA | 152 | Current, ratio of amperages. Note: Users may need to supply a prefix such as ‘m’ to show rates such as ‘mA/A’. |
| VPerVA | 153 | Power factor, PF, the ratio of the active power to the apparent power. Note: The sign convention used for power factor will differ between IEC meters and EEI (ANSI) meters. It is assumed that the data consumers understand the type of meter being used and agree on the sign convention in use at any given utility. |
| rev | 154 | Amount of rotation, revolutions. |
| kat | 158 | Catalytic activity, katal = mol / s. |
| JPerkg | 165 | Specific energy, Joules / kg. |
| m3Uncompensated | 166 | Volume, cubic metres, with the value uncompensated for weather effects. |
| m3Compensated | 167 | Volume, cubic metres, with the value compensated for weather effects. |
| WPerW | 168 | Signal Strength, ratio of power. Note: Users may need to supply a prefix such as ‘m’ to show rates such as ‘mW/W’. |
| therm | 169 | Energy, therms. |
| onePerm | 173 | Wavenumber, reciprocal metres, (1/m). |
| m3Perkg | 174 | Specific volume, cubic metres per kilogram, v. |
| Pas | 175 | Dynamic viscosity, pascal seconds. |
| Nm | 176 | Moment of force, newton metres. |
| NPerm | 177 | Surface tension, newton per metre. |
| radPers2 | 178 | Angular acceleration, radians per second squared. |
| JPerm3 | 181 | Energy density, joules per cubic metre. |
| VPerm | 182 | Electric field strength, volts per metre. |
| CPerm3 | 183 | Electric charge density, coulombs per cubic metre. |
| CPerm2 | 184 | Surface charge density, coulombs per square metre. |
| FPerm | 185 | Permittivity, farads per metre. |
| HPerm | 186 | Permeability, henrys per metre. |
| JPermol | 187 | Molar energy, joules per mole. |
| JPermolK | 188 | Molar entropy, molar heat capacity, joules per mole kelvin. |
| CPerkg | 189 | Exposure (x rays), coulombs per kilogram. |
| GyPers | 190 | Absorbed dose rate, grays per second. |
| WPersr | 191 | Radiant intensity, watts per steradian. |
| WPerm2sr | 192 | Radiance, watts per square metre steradian. |
| katPerm3 | 193 | Catalytic activity concentration, katals per cubic metre. |
| d | 195 | Time in days, day = 24 h = 86400 s. |
| anglemin | 196 | Plane angle, minutes. |
| anglesec | 197 | Plane angle, seconds. |
| ha | 198 | Area, hectares. |
| tonne | 199 | Mass in tons, “tonne” or “metric ton” (1000 kg = 1 Mg). |
| bar | 214 | Pressure in bars, (1 bar = 100 kPa). |
| mmHg | 215 | Pressure, millimetres of mercury (1 mmHg is approximately 133.3 Pa). |
| M | 217 | Length, nautical miles (1 M = 1852 m). |
| kn | 219 | Speed, knots (1 kn = 1852/3600) m/s. |
| Mx | 276 | Magnetic flux, maxwells (1 Mx = 10-8 Wb). |
| G | 277 | Magnetic flux density, gausses (1 G = 10-4 T). |
| Oe | 278 | Magnetic field in oersteds, (1 Oe = (103/4p) A/m). |
| Vh | 280 | Volt-hour, Volt hours. |
| WPerA |  | Active power per current flow, watts per Ampere. |
| onePerHz |  | Reciprocal of frequency (1/Hz). |
| VPerVAr |  | Power factor, PF, the ratio of the active power to the apparent power. Note: The sign convention used for power factor will differ between IEC meters and EEI (ANSI) meters. It is assumed that the data consumers understand the type of meter being used and agree on the sign convention in use at any given utility. |
| ohmPerm | 86 | Electric resistance per length in ohms per metre ((V/A)/m). |
| kgPerJ |  | Weight per energy in kilograms per joule (kg/J). Note: multiplier “k” is included in this unit symbol for compatibility with IEC 61850-7-3. |
| JPers |  | Energy rate in joules per second (J/s). |

### AngleDegrees datatype

Measurement of angle in degrees.

Table 378 shows all attributes of AngleDegrees.

Table 378 – Attributes of DiagramLayoutProfile::AngleDegrees

| name | mult | type | description |
| --- | --- | --- | --- |
| value | 0..1 | [Float](#UML615) |  |
| unit | 0..1 | [UnitSymbol](#UML611) | (const=deg) |
| multiplier | 0..1 | [UnitMultiplier](#UML610) | (const=none) |

### Boolean primitive

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

### Date primitive

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

### Float primitive

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

### Integer primitive

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

### String primitive

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

# Common requirements and constraints

This subclause includes rules and constrains that are defined in IEC 61970-452, tagged "452". They are included to make the validation self-contained. The rule and constraints that are tagged "457" are mastered in this document.

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

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

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

The string IdentifiedObject.description is maximum of 256 characters.

* R:452:ALL:NA:float

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

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

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

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

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

* R:452:ALL:NA:uniqueIdentifier

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

* R:452:ALL:NA:exchange

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

* R:452:ALL:NA:exchange1

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

* R:452:ALL:NA:exchange2

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

* R:452:ALL:NA:exchange3

An exporter may, at his or her discretion, produce a serialization containing additional class data described by the CIM Schema but not required by this document provided these data adhere to the conventions established in Clause 8.

1. (normative)  
     
   xxx

xxx

1. (informative)  
     
   Examples using IEC 61970-552 serialisation (placeholder in case we want to include something like this)
   1. Overview

The purpose of this interface is to provide the dynamics power system models for generators, loads, and any other equipment which may be needed for small signal stability studies of a power system. The standard interconnections diagram in **Error! Reference source not found.** along with the UML in **Error! Reference source not found.** shall be used in this example. In this example, a generator dynamics model and its parameters are specified. Figure B.1 shows an example of the header portion of an exchange containing a Dynamics payload. The uri specified in the Model.DependsOn attribute is the uri of the equipment model which this dynamics’ model references. Model Authority sets may also be specified if different authorities are contributing data to the model.

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:cim="http://iec.ch/TC57/CIM100#" xmlns:md="http://iec.ch/TC57/61970-552/ModelDescription/1#">

<md:FullModel rdf:about="urn:uuid:dc8d2d28-89cf-11e3-8aa9-2c44fd6790f2">

<md:Model.created>2014-01-31T12:09:44Z</md:Model.created>

<md:Model.version>1</md:Model.version>

<md:Model.description />

<md:Model.profile> http://iec.ch/TC57/CIM/Dynamics/1/0</md:Model.profile>

<md:Model.DependentOn rdf:resource="urn:uuid:caca0418-89ce-11e3-8aa9-2c44fd6790f2" />

</md:FullModel>

Figure B.1 – Dynamics model header

* 1. xxx

Bibliography (to be updated)

IEC 61400-27-1:2015, Wind Turbines – Part 27-1: – Electrical simulation models – Wind turbines

IEC 61400-27-1:2020, Wind Turbines – Part 27-1: – Electrical simulation models – Wind turbines

IEC 61970-552:2016, Energy management system application program interface (EMS-API) – Part 552: CIMXML model exchange formatIEC 61970-501:2006, Energy management system application program interface (EMS-API) – Part 501: Common information model resource description framework (CIM RDF) schema

IEEE 1110-1991, IEEE Guide for Synchronous Generator Practices in Stability Analysis

IEEE 421.5-2005, IEEE Recommended Practice for Excitation System Models for Power System Stability Studies

IEEE 421.5-2016, IEEE Recommended Practice for Excitation System Models for Power System Stability Studies

IEEE Task Force on Overall Plant Response, "Dynamic Models for Steam and Hydro Turbines in Power System Studies," IEEE Trans. PAS-92, Dec. 1973, pp. 1904-1915

IEEE Working Group on Prime Mover and Energy Supply Models, "Hydraulic Turbine and Turbine Control Models for System Dynamic Studies," IEEE Trans. on Power Systems, Vol. 7, No. 1, Feb. 1992, pp. 167-179

IEEE Special Publication, Dynamic Models for Turbine-Governors in Power System Studies, December 2012

IEEE/CIGRE, Stability Terms and Definitions

IEEE 421.2-1990, IEEE Guide for Identification, Testing, and Evaluation of the Dynamic Performance of Excitation Control Systems

IEEE PES-TR1, “Dynamic Models for Turbine-Governors in Power System Studies”, 2013

WECC REMWG, Proposal for new features for the renewable energy system generic models, 2021, https://www.wecc.org/Reliability/Memo\_RES\_Modeling\_Updates\_111121\_Rev22\_Clean.pdf

WECC ATSMWG, Proposal for hybrid STATCOM model (SVSMO4), 2021, <https://www.wecc.org/_layouts/15/WopiFrame.aspx?sourcedoc=/Administrative/Memo_SVSMO4_121621_Rev4.pdf&action=default&DefaultItemOpen=1>

\_\_\_\_\_\_\_\_\_\_\_

1. Under consideration. Stage at the time of publication: IEC/FDIS 61970-302:2022. [↑](#footnote-ref-2)