European Network of
Transmission System Operators
for Electricity

POWER SCHEDULE PROFILE SPECIFICATION

2024-10-16

ICTC APPROVED
VERSION 2.3.1

Copyright notice:

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

NOTE CONCERNING WORDING USED IN THIS DOCUMENT

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

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

Revision History

Version	Date	Paragraph	Comments
2.2.0	2023-03-24		For review.
2.2.0	2023-04-20		For ICTC approval.
2.3.0-alpha	2024-02-17		For internal review.
2.3.0-beta	2024-03-20		For CIM WG review.
2.3.1-alpha	2024-09-07		For CIM WG review.

CONTENTS

List of figures

List of tables

1 Introduction

The power schedule profile enables exchanging power schedule information and their time points.

2 Application profile specification

2.1 Version information

The content is generated from UML model file CIM17-2_CGMES31v01_PROF-20v02_NC23v65_MS10v01_DES10v01.eap.

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

- Title: Power schedule vocabulary
- Keyword: PS
- Description: This vocabulary is describing the object registry profile.
- Version IRI: https://ap-voc.cim4.eu/PowerSchedule/2.3
- Version info: 2.3.1
- Prior version: http://entsoe.eu/ns/CIM/PowerSchedule-EU/2.2
- Conforms to: urn:iso:std:iec:61970-600-2:ed-1|urn:iso:std:iec:61970-301:ed-7:amd1|file://iec61970cim17v40_iec61968cim13v13a_iec62325cim03v17a.eap|urn:iso:std:iec:61970-401:draft:ed-1|urn:iso:std:iec:61970-501:draft:ed-2|file://CIM100_CGMES31v01_501-20v02_NC23v62_MM10v01.eap
- Identifier: urn:uuid:470c9792-7798-4eb6-b7f2-6e18293c5f7b

2.2 Constraints naming convention

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

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

where

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

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

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

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

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

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

2.3 Profile constraints

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

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

- C:452:ALL:NA:datatypes

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

- R:452:ALL:NA:exchange

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

- R:452:ALL:NA:exchange1

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

- R:452:ALL:NA:exchange2

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

- R:452:ALL:NA:exchange3

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

- R:452:ALL:NA:exchange4

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

- R:452:ALL:NA:cardinality

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

- R:452:ALL:NA:associations

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

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

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

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

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

- R:452:ALL:NA:uniqueIdentifier

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

- R:452:ALL:NA:unitMultiplier

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

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

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

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

  The string IdentifiedObject.description is maximum 256 characters.

- C:452:ALL:NA:float

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

- R:NC:ALL:NA:serialization

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

1) the EAP file (EnterpriseArchitect project file),
2) the profiles’ specification document and
3) the application profiles (RDFS and SHACL).

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

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

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

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

- for association

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

- for attribute

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

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

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

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

- C:NC:PS:PowerSchedule:dc-associations

PowerSchedule can have associations to either DCTieCorridor or DCPole but not both.

2.4 Metadata

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

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

2.4.1 Constraints

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

- R:NC:ALL:wasAttributedTo:usage

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

2.4.2 Reference metadata

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

3 Detailed Profile Specification

3.1 General

This package contains the power schedule profile.

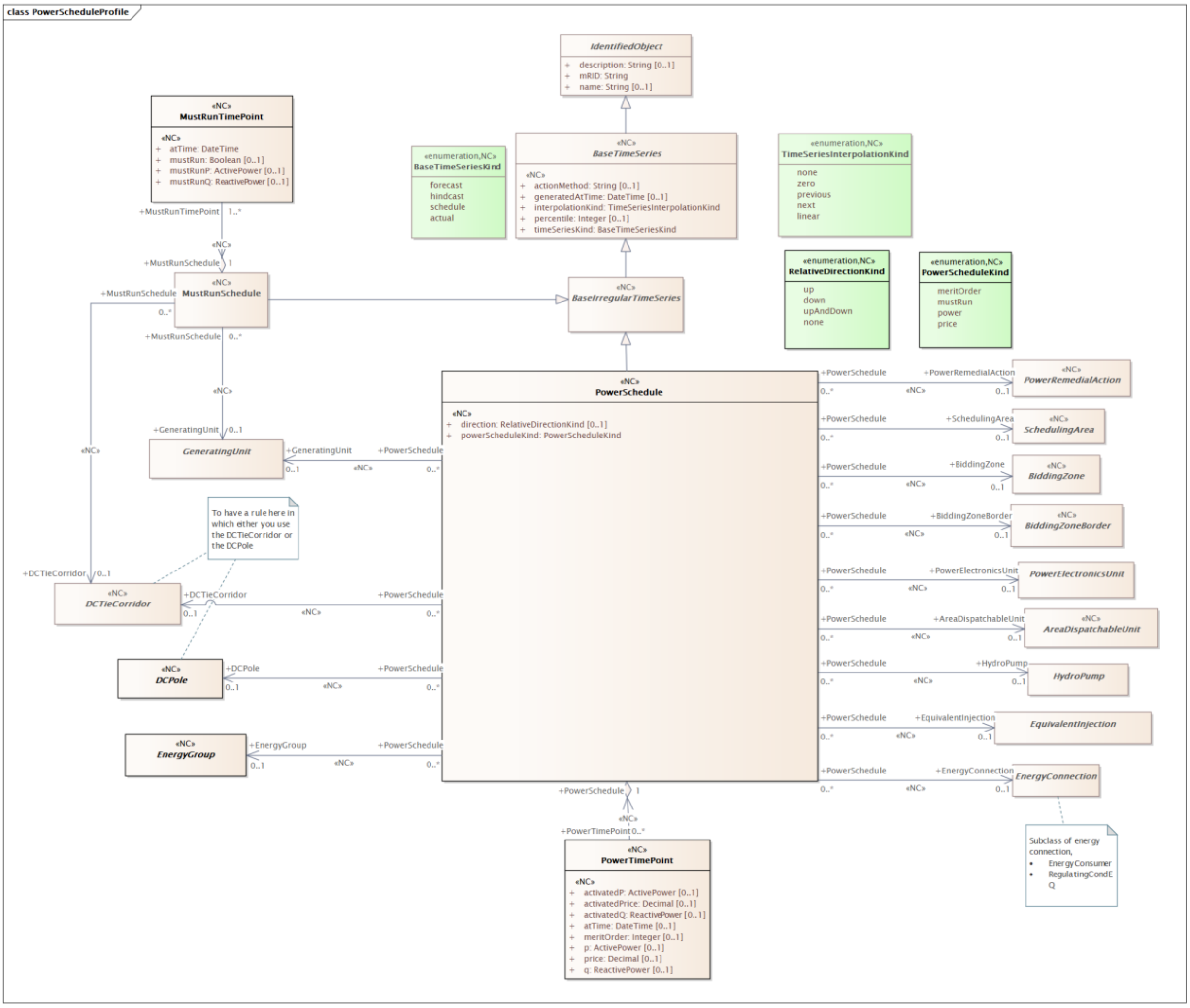

Figure 1 – Class diagram PowerScheduleProfile::PowerScheduleProfile

Figure 1: The diagram contains main classes related to the power schedule profile.

3.2 (abstract) IdentifiedObject root class

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

Table 1 shows all attributes of IdentifiedObject.

Table 1 – Attributes of PowerScheduleProfile::IdentifiedObject

name	mult	type	description
description	0..1	String	The description is a free human readable text describing or naming the object. It may be non unique and may not correlate to a naming hierarchy.
mRID	1..1	String	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

name	mult	type	description
			or rdf:about attributes that identify CIM object elements.
name	0..1	String	The name is any free human readable and possibly non unique text naming the object.

3.3 (abstract,NC) BaseTimeSeries

Inheritance path = IdentifiedObject
Time series of values at points in time.
Table 2 shows all attributes of BaseTimeSeries.

Table 2 – Attributes of PowerScheduleProfile::BaseTimeSeries

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	(NC) Kind of interpolation done between time point.
timeSeriesKind	1..1	BaseTimeSeriesKind	(NC) Kind of base time series.
generatedAtTime	0..1	DateTime	(NC) The time this time series (entity) come to existents and available for use.
percentile	0..1	Integer	(NC) The percentile is a number where a certain percentage of scores/ranking/values of a sample fall below that number. This is a way for expressing uncertainty in the number provided.
actionMethod	0..1	String	(NC) Action method used to create the value. This is used for identification in the case where there is multiple time series for the same validity period and kind.
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

3.4 (abstract,NC) BaseIrregularTimeSeries

Inheritance path = BaseTimeSeries : IdentifiedObject
Time series that has irregular points in time.
Table 3 shows all attributes of BaseIrregularTimeSeries.

Table 3 – Attributes of PowerScheduleProfile::BaseIrregularTimeSeries

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	(NC) inherited from: BaseTimeSeries
timeSeriesKind	1..1	BaseTimeSeriesKind	(NC) inherited from: BaseTimeSeries
generatedAtTime	0..1	DateTime	(NC) inherited from: BaseTimeSeries
percentile	0..1	Integer	(NC) inherited from: BaseTimeSeries
actionMethod	0..1	String	(NC) inherited from: BaseTimeSeries
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

3.5 (NC) BaseTimeSeriesKind enumeration

Kind of time series.

Table 4 shows all literals of BaseTimeSeriesKind.

Table 4 – Literals of PowerScheduleProfile::BaseTimeSeriesKind

literal	value	description
forecast		Time series is forecast data. The values represent the result of scientific predictions based on historical time stamped data.
hindcast		Time series is hindcast data. The value represent probable past (historic) condition given by calculation done using actual values. For instance, determine the among of wind based on the energy produced by wind. However, hindcast is typical the result of a simulated forecasts for historical periods.
schedule		Time series is schedule data. The values represent the result of a committed and plan forecast data that has been through a quality control and could incur penalty when not followed.
actual		Time series is actual data. The values represent measured or calculated values that represent the actual behaviour.

3.6 (NC) TimeSeriesInterpolationKind enumeration

Kinds of interpolation of values between two time point.
Table 5 shows all literals of TimeSeriesInterpolationKind.

Table 5 – Literals of PowerScheduleProfile::TimeSeriesInterpolationKind

literal	value	description
none		No interpolation is applied.
zero		The value between two time points is set to zero.
previous		The value between two time points is set to previous value.
next		The value between two time points is set to next value.
linear		Linear interpolation is applied for values between two time points.

3.7 (NC) PowerSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
Time series represent irregular power, active and reactive, values at given points in time.
Table 6 shows all attributes of PowerSchedule.

Table 6 – Attributes of PowerScheduleProfile::PowerSchedule

name	mult	type	description
direction	0..1	RelativeDirectionKind	(NC) Kind of direction.
powerScheduleKind	1..1	PowerScheduleKind	(NC) Kind of power schedule.
interpolationKind	1..1	TimeSeriesInterpolation Kind	(NC) inherited from: BaseTimeSeries
timeSeriesKind	1..1	BaseTimeSeriesKind	(NC) inherited from: BaseTimeSeries
generatedAtTime	0..1	DateTime	(NC) inherited from: BaseTimeSeries
percentile	0..1	Integer	(NC) inherited from: BaseTimeSeries

name	mult	type	description
actionMethod	0..1	String	(NC) inherited from: BaseTimeSeries
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

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

Table 7 – Association ends of PowerScheduleProfile::PowerSchedule with other classes

mult from	name	mult to	type	description
0..*	EnergyConnection	0..1	EnergyConnection	(NC) The energy connection that has a power schedule.
0..*	SchedulingArea	0..1	SchedulingArea	Scheduling area which has power schedules.
0..*	AreaDispatchableUnit	0..1	AreaDispatchableUnit	(NC) Area disptachable unit which belongs to the power schedule.
0..*	DCTieCorridor	0..1	DCTieCorridor	(NC) DC tie corridor which belongs to the power schedule.
0..*	GeneratingUnit	0..1	GeneratingUnit	(NC) Generating unit which belongs to the power schedule.
0..*	HydroPump	0..1	HydroPump	(NC) Hydro pump which belongs to the power schedule.
0..*	PowerElectronicsUnit	0..1	PowerElectronicsUnit	(NC) Power electronics unit which belongs to the power schedule.
0..*	PowerRemedialAction	0..1	PowerRemedialAction	(NC) Power remedial action which belongs to the Remedial Action Schedule.
0..*	BiddingZoneBorder	0..1	BiddingZoneBorder	(NC) Bidding zone border which belongs to the power schedule.
0..*	EquivalentInjection	0..1	EquivalentInjection	(NC) Equivalent injection which belongs to the power schedule.
0..*	DCPole	0..1	DCPole	(NC) DC pole which belongs to the power schedule.
0..*	EnergyGroup	0..1	EnergyGroup	(NC) Energy group which belongs to a power schedule.
0..*	BiddingZone	0..1	BiddingZone	(NC) Bidding zone which has power schedules.

3.8 (NC) PowerTimePoint root class

Power, active and reactive, value at a given point in time.
Table 8 shows all attributes of PowerTimePoint.

Table 8 – Attributes of PowerScheduleProfile::PowerTimePoint

name	mult	type	description
atTime	0..1	DateTime	(NC) The time the data is valid for.
p	0..1	ActivePower	(NC) Active power injection. Load sign convention is used, i.e. positive sign means flow out from a node.

name	mult	type	description
q	0..1	ReactivePower	(NC) Reactive power injection. Load sign convention is used, i.e. positive sign means flow out from a node.
price	0..1	Decimal	(NC) Price for the scheduled active power per unit of active power. e.g. per MW.
activatedP	0..1	ActivePower	(NC) Active power activated as part of redispatch. Negative number means that the value is scheduling down. Positive number means that the value is scheduling up.
activatedPrice	0..1	Decimal	(NC) Price for the activated active power per unit e.g. per MW.
activatedQ	0..1	ReactivePower	(NC) Reactive power activated as part of redispatch. Negative number means that the value is scheduling down. Positive number means that the value is scheduling up.
meritOrder	0..1	Integer	(NC) Ranking the energy blocks. Ranking can be based on historical values or other sources. It is required if power schedule is kind merit order.

Table 9 shows all association ends of PowerTimePoint with other classes.

Table 9 – Association ends of PowerScheduleProfile::PowerTimePoint with other classes

mult from	name	mult to	type	description
0..*	PowerSchedule	1..1	PowerSchedule	(NC) Time series the time point values belongs to.

3.9 (NC) MustRunSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
Time series represent irregular must-run instruction values at given points in time. This could be instruction to a reliability must-run (RMR) generation facility that is necessary to run to meet certain operating conditions in order to maintain the security of power systems in a competitive environment.
Table 10 shows all attributes of MustRunSchedule.

Table 10 – Attributes of PowerScheduleProfile::MustRunSchedule

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	(NC) inherited from: BaseTimeSeries
timeSeriesKind	1..1	BaseTimeSeriesKind	(NC) inherited from: BaseTimeSeries
generatedAtTime	0..1	DateTime	(NC) inherited from: BaseTimeSeries
percentile	0..1	Integer	(NC) inherited from: BaseTimeSeries
actionMethod	0..1	String	(NC) inherited from: BaseTimeSeries
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

Table 11 shows all association ends of MustRunSchedule with other classes.

Table 11 – Association ends of PowerScheduleProfile::MustRunSchedule with other classes

mult from	name	mult to	type	description
0..*	DCTieCorridor	0..1	DCTieCorridor	(NC) Hydro pump which belongs to the power schedule.
0..*	GeneratingUnit	0..1	GeneratingUnit	(NC) Generating unit which belongs to the must run schedule.

3.10 (NC) MustRunTimePoint root class

Must-run instruction value at a given point in time.

Table 12 shows all attributes of MustRunTimePoint.

Table 12 – Attributes of PowerScheduleProfile::MustRunTimePoint

name	mult	type	description
atTime	1..1	DateTime	(NC) The time the data is valid for.
mustRunP	0..1	ActivePower	(NC) Minimum active power injection that is needed to meet must-run requirement. This value can be higher or equal to minimum operational limit. Load sign convention is used, i.e. positive sign means flow out from a node.
mustRunQ	0..1	ReactivePower	(NC) Minimum reactive power injection that is needed to meet must-run requirement. This value can be higher or equal to minimum operational limit. Load sign convention is used, i.e. positive sign means flow out from a node.
mustRun	0..1	Boolean	(NC) True, if the must-run instruction is active this time point. Otherwise false.

Table 13 shows all association ends of MustRunTimePoint with other classes.

Table 13 – Association ends of PowerScheduleProfile::MustRunTimePoint with other classes

mult from	name	mult to	type	description
1..*	MustRunSchedule	1..1	MustRunSchedule	(NC) Time series the time point values belongs to.

3.11 (abstract) EnergyConnection root class

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

3.12 (abstract) HydroPump root class

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

3.13 (abstract,NC) AreaDispatchableUnit root class

Allocates a given producing or consuming unit, including direct current corridor and collection of units, to a given control area (through the scheduling area) for supporting the control of the given area through dispatch instruction.

3.14 (abstract) GeneratingUnit root class

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.

3.15 (abstract) PowerElectronicsUnit root class

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

3.16 (abstract,NC) DCTieCorridor root class

A collection of one or more direct current poles that connect two different control areas.

3.17 (abstract,NC) BiddingZone root class

A bidding zone is a market-based method for handling power transmission congestion. It consists of scheduling areas that include the relevant production (supply) and consumption (demand) to form an electrical area with the same market price without capacity allocation.

3.18 (abstract,NC) SchedulingArea root class

An area where production and/or consumption of energy can be forecasted, scheduled and measured. The area is operated by only one system operator, typically a Transmission System Operator (TSO). The area can consist of a sub area, which has the same definition as the main area, but it can be operated by another system operator (typically Distributed System Operator (DSO) or a Closed Distributed System Operator (CDSO)). This includes microgrid concept. A substation is the smallest grouping that can be included in the area. The area size should be considered in terms of the possibility of accumulated reading (settlement metering) and the capability of operating as an island.

3.19 DateTime primitive

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

3.20 Integer primitive

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

3.21 String primitive

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

3.22 Boolean primitive

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

3.23 ActivePower datatype

Product of RMS value of the voltage and the RMS value of the in-phase component of the current.
Table 14 shows all attributes of ActivePower.

Table 14 – Attributes of PowerScheduleProfile::ActivePower

name	mult	type	description
multiplier	0..1	UnitMultiplier	(const=M)
unit	0..1	UnitSymbol	(const=W)
value	0..1	Float	

3.24 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 15 shows all literals of UnitMultiplier.

Table 15 – Literals of PowerScheduleProfile::UnitMultiplier

literal	value	description
M	6	Mega 10**6.

3.25 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 16 shows all literals of UnitSymbol.

Table 16 – Literals of PowerScheduleProfile::UnitSymbol

literal	value	description
W	38	Real power in watts (J/s). Electrical power may have real and reactive components. The real portion of electrical power (I^2R or VIcos(phi)), is expressed in Watts. See also apparent power and reactive power.
VAr	63	Reactive power in volt amperes reactive. The “reactive” or “imaginary” component of electrical

literal	value	description
		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.

3.26 Float primitive

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

3.27 ReactivePower datatype

Product of RMS value of the voltage and the RMS value of the quadrature component of the current.
Table 17 shows all attributes of ReactivePower.

Table 17 – Attributes of PowerScheduleProfile::ReactivePower

name	mult	type	description
value	0..1	Float	
unit	0..1	UnitSymbol	(const=VAr)
multiplier	0..1	UnitMultiplier	(const=M)

3.28 Decimal primitive

Decimal is the base-10 notational system for representing real numbers.

3.29 (abstract,NC) PowerRemedialAction root class

Energy remedial action describes actions to rearrange power schedules.

3.30 (abstract,NC) BiddingZoneBorder root class

Defines the aggregated connection capacity between two Bidding Zones.

3.31 (abstract) EquivalentInjection root class

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

3.32 (abstract,NC) DCPole root class

The direct current (DC) system pole (IEC 60633) is part of a DC system consisting of all the equipment in the DC substations and the interconnecting transmission lines, if any, which during normal operation exhibit a common direct voltage polarity with respect to earth.

3.33 (NC) RelativeDirectionKind enumeration

Kind of direction for the changes.
Table 18 shows all literals of RelativeDirectionKind.

Table 18 – Literals of PowerScheduleProfile::RelativeDirectionKind

literal	value	description
up		Up signifies that the changes are increasing from the current status.
down		Down signifies that the changes are decreasing from the current status.

literal	value	description
upAndDown		Up and down signifies that both up and down values are equal.
none		There is no direction on the changes.

3.34 (NC) PowerScheduleKind enumeration

Kind of power schedule.
Table 19 shows all literals of PowerScheduleKind.

Table 19 – Literals of PowerScheduleProfile::PowerScheduleKind

literal	value	description
meritOrder		Power schedule is a merit order that includes ranking of the power block. Power block provides the maximum power allocation possible.
mustRun		Power schedule is a must run schedule that identifies the unit that must run for a given time point.
power		Power schedule that has no additional meaning than allocating power to unit for a given time point.
price		Power schedule includes prices for a given a power per time point.

3.35 (abstract,NC) EnergyGroup root class

An energy group is an aggregation of energy components which have the same energy characteristic, e.g. fuel type and technology. It can be used to allocate energy.

Annex A (informative): Sample data

A.1 General

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

A.2 Sample instance data

Test data files are available in the CIM EG SharePoint.