European Network of
Transmission System Operators
for Electricity

AVAILABILITY 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.

33

Revision History

Version	Date	Paragraph	Comments
0.1.0	2021-10-12		For CIM EG review
1.0.0	2022-02-16		For SOC approval
2.1.0	2022-07-21		For SOC approval
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 availability schedule profile is a profile to exchange information on availability related to not only equipment, but also equipment containers, remedial action schemes, individual grid state alterations and collections of them, and operational limits. Availability schedules and functions are exchanged. A given (un)availability schedule provides information on status, cause and can include multiple equipment that is simultaneously scheduled for unavailability. The availability power system function gives the state change (e.g., inService, outOfService) of the relevant function (e.g., Line) for a single availability schedule. Only power system functions that are directly impacted are explicitly included. For example, the unavailability of a switch might cause a line to be unavailable. Only the switch is included in the schedule and not the line that becomes de-energized as a cause of the availability schedule for switch.

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: Availability schedule vocabulary
- Keyword: AS
- Description: This vocabulary is describing the availability schedule profile.
- Version IRI: https://ap-voc.cim4.eu/AvailabilitySchedule/2.3
- Version info: 2.3.1
- Prior version: http://entsoe.eu/ns/CIM/AvailabilitySchedule-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:8d128e35-86c7-4d67-b2dd-93229bf1005a

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:AS:AvailabilityEnabled:associations

  AvailabilityEnabled shall have an association with either GridStateAlteration or AssessedElement.

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 availability schedule profile.

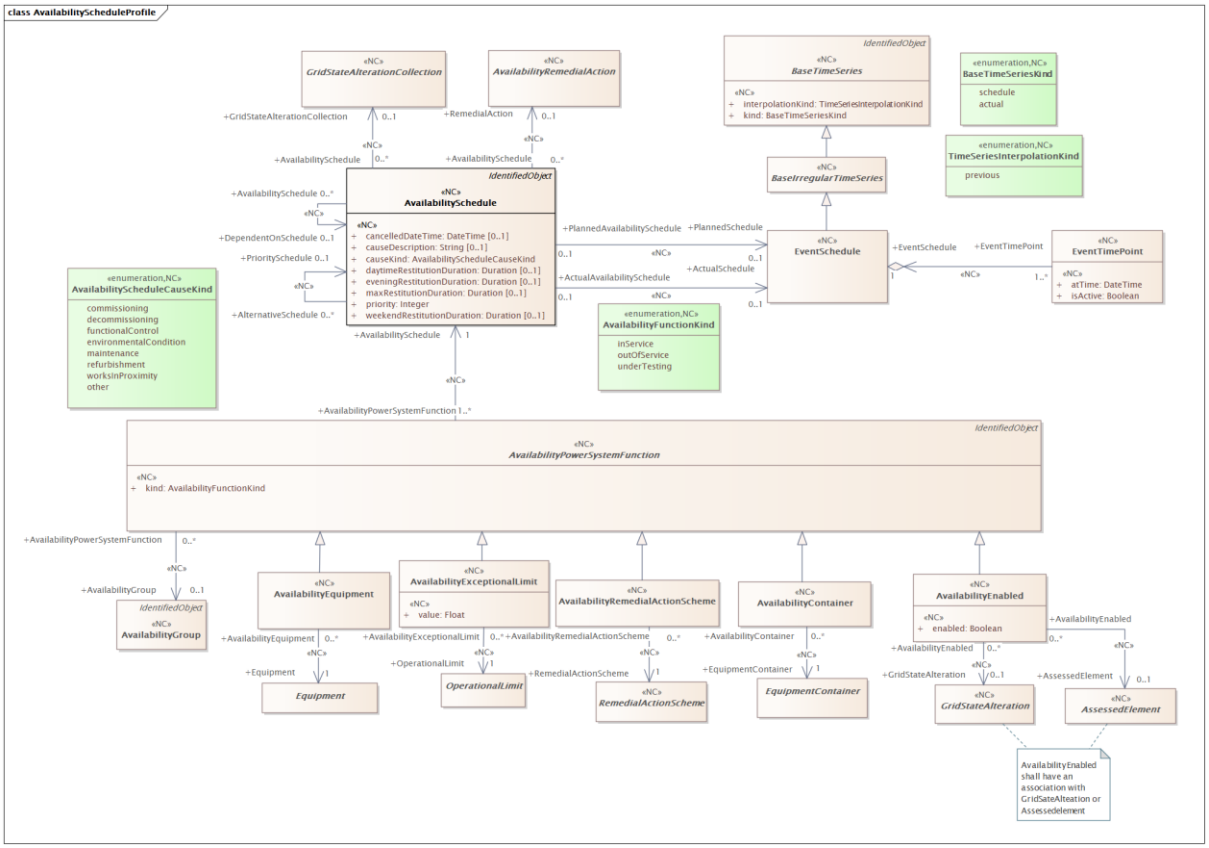

Figure 1 – Class diagram AvailabilityScheduleProfile::AvailabilityScheduleProfile

Figure 1: The diagram contains the main classes used in the availability schedule profile.

3.2 (abstract,NC) AvailabilityRemedialAction root class

Availability remedial action is a remedial action that cancels or reschedules an availability schedule.

3.3 (NC) EventSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
Time series represent irregular event described by event points in time.
Table 1 shows all attributes of EventSchedule.

Table 1 – Attributes of AvailabilityScheduleProfile::EventSchedule

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	(NC) inherited from: BaseTimeSeries
kind	1..1	BaseTimeSeriesKind	(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

3.4 (NC) EventTimePoint root class

Event valid for a given point in time.
Table 2 shows all attributes of EventTimePoint.

Table 2 – Attributes of AvailabilityScheduleProfile::EventTimePoint

name	mult	type	description
atTime	1..1	DateTime	(NC) The time the data is valid for.
isActive	1..1	Boolean	(NC) True, if the event is occurring (Active) at this time point. Otherwise false.

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

Table 3 – Association ends of AvailabilityScheduleProfile::EventTimePoint with other classes

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

3.5 (abstract,NC) GridStateAlterationCollection root class

A collection of grid state alterations.

3.6 (NC) AvailabilityEquipment

Inheritance path = AvailabilityPowerSystemFunction : IdentifiedObject

Availability equipment serves for associating an equipment with an availability schedule. For instance, putting in or out of service an ACLineSegment in combination with other availability functions with the same availability schedule.

Table 4 shows all attributes of AvailabilityEquipment.

Table 4 – Attributes of AvailabilityScheduleProfile::AvailabilityEquipment

name	mult	type	description
kind	1..1	AvailabilityFunctionKind	(NC) inherited from: AvailabilityPowerSystemFunction
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

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

Table 5 – Association ends of AvailabilityScheduleProfile::AvailabilityEquipment with other classes

mult from	name	mult to	type	description
0..*	Equipment	1..1	Equipment	(NC) Equipment that is affected by the availability given by this availability equipment.
0..*	AvailabilityGroup	0..1	AvailabilityGroup	(NC) inherited from: AvailabilityPowerSystemFunction
1..*	AvailabilitySchedule	1..1	AvailabilitySchedule	(NC) inherited from: AvailabilityPowerSystemFunction

3.7 (NC) AvailabilityExceptionalLimit

Inheritance path = AvailabilityPowerSystemFunction : IdentifiedObject

Availability exceptional limit serves for associating an operational limit restriction with an availability schedule. For instance, enabling or disabling the current limit on ACLineSegment terminal in combination with other availability functions with the same availability schedule or de-rating due to fault.
Table 6 shows all attributes of AvailabilityExceptionalLimit.

Table 6 – Attributes of AvailabilityScheduleProfile::AvailabilityExceptionalLimit

name	mult	type	description
value	1..1	Float	(NC) Value for the referred operational limit.
kind	1..1	AvailabilityFunctionKind	(NC) inherited from: AvailabilityPowerSystemFunction
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 AvailabilityExceptionalLimit with other classes.

Table 7 – Association ends of AvailabilityScheduleProfile::AvailabilityExceptionalLimit with other classes

mult from	name	mult to	type	description
0..*	OperationalLimit	1..1	OperationalLimit	(NC) Operational limit that is constrained by this availability exceptional limit.
0..*	AvailabilityGroup	0..1	AvailabilityGroup	(NC) inherited from: AvailabilityPowerSystemFunction
1..*	AvailabilitySchedule	1..1	AvailabilitySchedule	(NC) inherited from: AvailabilityPowerSystemFunction

3.8 (NC) AvailabilityGroup

Inheritance path = IdentifiedObject
Container to link relevant equipment that is affected by (un)availability schedule across availability coordinator (e.g. TSO-TSO, TSO-DSO or DSO-DSO).
Table 8 shows all attributes of AvailabilityGroup.

Table 8 – Attributes of AvailabilityScheduleProfile::AvailabilityGroup

name	mult	type	description
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

3.9 (abstract,NC) AvailabilityPowerSystemFunction

Inheritance path = IdentifiedObject
Availability power system function describes the power system function that has a non-normal availability in the associated availability schedule. The availability of the function is needed as part of a power flow solution. This function is the cause and not the effect of the availability, if the effect can be calculated through power flow. For instance if only the step-up transformer for a generator is not available, the power flow will calculate that the generator is de-energized (outage). If both are tagged as not available it will not be possible to investigate remedial action for connecting the generator. It is expected that the power flow function is able to perform simple topology changes affected by a function taken out of service, e.g. open switches on both end

of a ACLineSegment when the ACLineSegment is taken out of service. More complex changes, like change regulation set point, must be described in the linked GridStateAlterationCollection. Table 9 shows all attributes of AvailabilityPowerSystemFunction.

Table 9 – Attributes of AvailabilityScheduleProfile::AvailabilityPowerSystemFunction

name	mult	type	description
kind	1..1	AvailabilityFunctionKind	(NC) Kind of availability that affect the power system function.
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

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

Table 10 – Association ends of AvailabilityScheduleProfile::AvailabilityPowerSystemFunction with other classes

mult from	name	mult to	type	description
0..*	AvailabilityGroup	0..1	AvailabilityGroup	(NC) Grouping for all availability power system functions (controlled by all relevant system operators) that have the same availability schedule.
1..*	AvailabilitySchedule	1..1	AvailabilitySchedule	(NC) Availability schedule for this availability power system function.

3.10 (NC) AvailabilityRemedialActionScheme

Inheritance path = AvailabilityPowerSystemFunction : IdentifiedObject

Availability remedial action scheme serves for associating a remedial action scheme with an availability schedule. For instance, taking in or out of service a SIPS / SPS due to communication issue, in combination with other availability functions with the same availability schedule.

Table 11 shows all attributes of AvailabilityRemedialActionScheme.

Table 11 – Attributes of AvailabilityScheduleProfile::AvailabilityRemedialActionScheme

name	mult	type	description
kind	1..1	AvailabilityFunctionKind	(NC) inherited from: AvailabilityPowerSystemFunction
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

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

Table 12 – Association ends of AvailabilityScheduleProfile::AvailabilityRemedialActionScheme with other classes

mult from	name	mult to	type	description
0..*	RemedialActionScheme	1..1	RemedialActionScheme	(NC) Remedial action scheme that is affected by the availability given by this availability remedial action scheme.

mult from	name	mult to	type	description
0..*	AvailabilityGroup	0..1	AvailabilityGroup	(NC) inherited from: AvailabilityPowerSystemFunction
1..*	AvailabilitySchedule	1..1	AvailabilitySchedule	(NC) inherited from: AvailabilityPowerSystemFunction

3.11 (NC) AvailabilitySchedule

Inheritance path = IdentifiedObject

A given (un)availability schedule with a given status and cause that include multiple equipment that need to follow the same scheduling periods.

Table 13 shows all attributes of AvailabilitySchedule.

Table 13 – Attributes of AvailabilityScheduleProfile::AvailabilitySchedule

name	mult	type	description
cancelledDateTime	0..1	DateTime	(NC) The date and time the (un)availability schedule were cancelled .
causeDescription	0..1	String	(NC) A cause description for a cause kind. In case of CauseKind equals other, description or a reference of the cause of the (un)availability schedule.
causeKind	1..1	AvailabilityScheduleCauseKind	(NC) Kind of cause for the availability schedule.
maxRestitutionDuration	0..1	Duration	(NC) The maximum time required to take the out-of-service equipment back into service. This includes the start-up time for generating units.
priority	1..1	Integer	(NC) Value 0 means ignore priority. 1 means the highest priority, 2 is the second highest priority.
daytimeRestitutionDuration	0..1	Duration	(NC) The time required to take the out-of-service equipment back into service during daytime. This includes the start-up time for generating units.
eveningRestitutionDuration	0..1	Duration	(NC) The time required to take the out-of-service equipment back into service after office hours. This includes the start-up time for generating units.
weekendRestitutionDuration	0..1	Duration	(NC) The time required to take the out-of-service equipment back into service in the weekend or during bank holidays. This includes the start-up time for generating units.
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

Table 14 shows all association ends of AvailabilitySchedule with other classes.

Table 14 – Association ends of AvailabilityScheduleProfile::AvailabilitySchedule with other classes

mult from	name	mult to	type	description
0..*	RemedialAction	0..1	AvailabilityRemedialAction	(NC) Remedial action that is cancelling this availability schedule.
0..*	GridStateAlterationCollection	0..1	GridStateAlterationCollection	(NC) The grid state alteration collection that has this availability schedule.

mult from	name	mult to	type	description
0..*	DependentOnSchedule	0..1	AvailabilitySchedule	(NC) (Un)availability schedule requested by one operator may require another operator to request their (un)availability schedule. This association is linking the schedules so that the dependency is clear.
0..1	ActualSchedule	0..1	EventSchedule	(NC) Actual schedule that relates to this availability schedule; used for ex-post reporting and analysis (e.g., to compare planned vs. actual).
0..1	PlannedSchedule	0..1	EventSchedule	(NC) Planned schedule that relates to this availability schedule used for planning availability (e.g., to compare planned vs. actual).
0..*	PrioritySchedule	0..1	AvailabilitySchedule	(NC) Priority schedule. This is the schedule that has the highest priority and the only valid if not cancelled.

3.12 (NC) AvailabilityContainer

Inheritance path = AvailabilityPowerSystemFunction : IdentifiedObject
Availability container serves for associating an equipment container with an availability schedule. For instance, putting in or out of service all the equipment inside a Line or a Bay in combination with other availability functions with the same availability schedule.
Table 15 shows all attributes of AvailabilityContainer.

Table 15 – Attributes of AvailabilityScheduleProfile::AvailabilityContainer

name	mult	type	description
kind	1..1	AvailabilityFunctionKind	(NC) inherited from: AvailabilityPowerSystemFunction
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

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

Table 16 – Association ends of AvailabilityScheduleProfile::AvailabilityContainer with other classes

mult from	name	mult to	type	description
0..*	EquipmentContainer	1..1	EquipmentContainer	(NC) Equipment container that is affected by the availability given by this availability container.
0..*	AvailabilityGroup	0..1	AvailabilityGroup	(NC) inherited from: AvailabilityPowerSystemFunction
1..*	AvailabilitySchedule	1..1	AvailabilitySchedule	(NC) inherited from: AvailabilityPowerSystemFunction

3.13 (abstract,NC) BaseIrregularTimeSeries

Inheritance path = BaseTimeSeries : IdentifiedObject
Time series that has irregular points in time.
Table 17 shows all attributes of BaseIrregularTimeSeries.

Table 17 – Attributes of AvailabilityScheduleProfile::BaseIrregularTimeSeries

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	(NC) inherited from: BaseTimeSeries
kind	1..1	BaseTimeSeriesKind	(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

3.14 (abstract,NC) BaseTimeSeries

Inheritance path = IdentifiedObject

Time series of values at points in time.

Table 18 shows all attributes of BaseTimeSeries.

Table 18 – Attributes of AvailabilityScheduleProfile::BaseTimeSeries

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	(NC) Kind of interpolation done between time point.
kind	1..1	BaseTimeSeriesKind	(NC) Kind of base time series.
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

3.15 (abstract) Equipment root class

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

3.16 (abstract) EquipmentContainer root class

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

3.17 (abstract,NC) GridStateAlteration root class

Grid state alteration is a change of values describing state (operating point) of one element in the grid model compared to the base case.

3.18 (abstract) IdentifiedObject root class

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

Table 19 shows all attributes of IdentifiedObject.

Table 19 – Attributes of AvailabilityScheduleProfile::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.

name	mult	type	description
			For CIMXML data files in RDF syntax conforming to IEC 61970-552, the mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements.
name	0..1	String	The name is any free human readable and possibly non unique text naming the object.

3.19 (abstract) OperationalLimit root class

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.

3.20 (abstract,NC) RemedialActionScheme root class

Remedial Action Scheme (RAS), Special Protection Schemes (SPS), System Protection Schemes (SPS) or System Integrity Protection Schemes (SIPS).
A Remedial Action Scheme consists of one or more stages that can trigger and execute a protection action.

3.21 (NC) TimeSeriesInterpolationKind enumeration

Kinds of interpolation of values between two time point.
Table 20 shows all literals of TimeSeriesInterpolationKind.

Table 20 – Literals of AvailabilityScheduleProfile::TimeSeriesInterpolationKind

literal	value	description
previous		The value between two time points is set to previous value.

3.22 (NC) AvailabilityFunctionKind enumeration

Kind of availability that is affecting the function.
Table 21 shows all literals of AvailabilityFunctionKind.

Table 21 – Literals of AvailabilityScheduleProfile::AvailabilityFunctionKind

literal	value	description
inService		Function is in service.
outOfService		Function is out-of-service.
underTesting		Function is under testing and need to expect unscheduled availability.

3.23 (NC) AvailabilityScheduleCauseKind enumeration

The kinds of cause of the (un)availability schedule.
Table 22 shows all literals of AvailabilityScheduleCauseKind.

Table 22 – Literals of AvailabilityScheduleProfile::AvailabilityScheduleCauseKind

literal	value	description
commissioning		The cause is due to a commissioning.

literal	value	description
decommissioning		The cause is due to a decommissioning.
functionalControl		The cause is due to a functional control (in & out).
environmentalCondition		The cause is due to an environmental condition. This can lead to exceptional margin and limits.
maintenance		The cause is due to a maintenance.
refurbishment		The cause is due to a refurbishment, either upgrade or downgrade.
worksInProximity		The cause is due to a works in proximity.
other		The cause is of other kind.

3.24 (NC) BaseTimeSeriesKind enumeration

Kind of time series.
Table 23 shows all literals of BaseTimeSeriesKind.

Table 23 – Literals of AvailabilityScheduleProfile::BaseTimeSeriesKind

literal	value	description
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.

3.25 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 24 shows all literals of UnitMultiplier.

Table 24 – Literals of AvailabilityScheduleProfile::UnitMultiplier

literal	value	description
none	0	No multiplier or equivalently multiply by 1.

3.26 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 25 shows all literals of UnitSymbol.

Table 25 – Literals of AvailabilityScheduleProfile::UnitSymbol

literal	value	description
s	4	Time in seconds.

3.27 Seconds datatype

Time, in seconds.
Table 26 shows all attributes of Seconds.

Table 26 – Attributes of AvailabilityScheduleProfile::Seconds

name	mult	type	description
value	0..1	Float	Time, in seconds
unit	0..1	UnitSymbol	(const=s)
multiplier	0..1	UnitMultiplier	(const=none)

3.28 Boolean primitive

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

3.29 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.

3.30 Duration primitive

Duration as "PnYnMnDTnHnMnS" which conforms to ISO 8601, where nY expresses a number of years, nM a number of months, nD a number of days. The letter T separates the date expression from the time expression and, after it, nH identifies a number of hours, nM a number

of minutes and nS a number of seconds. The number of seconds could be expressed as a decimal number, but all other numbers are integers.

3.31 Integer primitive

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

3.32 Float primitive

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

3.33 String primitive

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

3.34 (NC) AvailabilityEnabled

Inheritance path = AvailabilityPowerSystemFunction : IdentifiedObject
Availability enabled is enabling or disabling grid state alteration (e.g. tap position action) or assessed element that is related to the availability schedule. For instance, the cancellation of availability schedule can lead to changes in the assessed element. This is done by enabling one assessment and disabling another.
Table 27 shows all attributes of AvailabilityEnabled.

Table 27 – Attributes of AvailabilityScheduleProfile::AvailabilityEnabled

name	mult	type	description
enabled	1..1	Boolean	(NC) Instruction to enable or disable alteration and assessment.
kind	1..1	AvailabilityFunctionKind	(NC) inherited from: AvailabilityPowerSystemFunction
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

Table 28 shows all association ends of AvailabilityEnabled with other classes.

Table 28 – Association ends of AvailabilityScheduleProfile::AvailabilityEnabled with other classes

mult from	name	mult to	type	description
0..*	GridStateAlteration	0..1	GridStateAlteration	(NC) Grid state alteration that is affected by the availability given by this availability enabling.
0..*	AssessedElement	0..1	AssessedElement	(NC) Assessed element that is affected by the availability given by this availability enabling.
0..*	AvailabilityGroup	0..1	AvailabilityGroup	(NC) inherited from: AvailabilityPowerSystemFunction
1..*	AvailabilitySchedule	1..1	AvailabilitySchedule	(NC) inherited from: AvailabilityPowerSystemFunction

3.35 (abstract,NC) AssessedElement root class

Assessed element is a network element for which the electrical state is evaluated in the regional or cross-regional process and which value is expected to fulfil regional rules function of the operational security limits.
The measurements and limits are as defined in the steady state hypothesis.

589

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.