European Network of
Transmission System Operators
for Electricity

AVAILABILITY SCHEDULE PROFILE SPECIFICATION

2022-09-21

SOC APPROVED
VERSION 2.1

Copyright notice:

This document is maintained by the ENTSO-E CIM EG. 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	Release	Date	Paragraph	Comments
0	1	2021-10-12		For CIM EG review
1	0	2022-02-16		SOC approved.
2	1	2022-09-21		SOC approved.

CONTENTS

List of figures

List of tables

1 Introduction

The availability schedule profile is a profile to exchange information on availability related to different equipment. 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 is an instruction on the relevant power system function availability in regard to a given 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 CIM100_CGMES31v01_501-20v02_NC21v52_MM10v01.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: http://entsoe.eu/ns/CIM/AvailabilitySchedule-EU/2.1
- Version info: 2.1.0
- Prior version:
- 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://CGMES-30v25_501-20v01.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.

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 (prov:Agent).

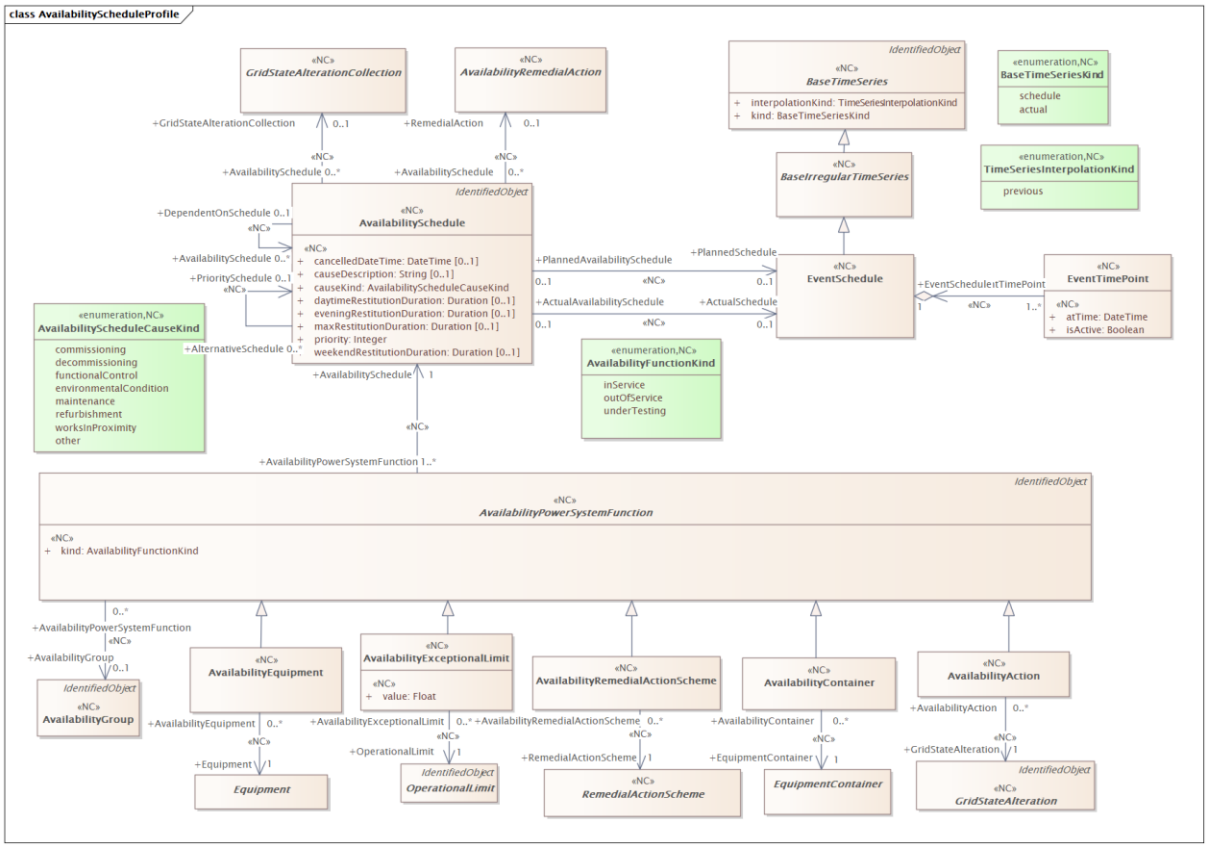
2.4.2 Reference metadata

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

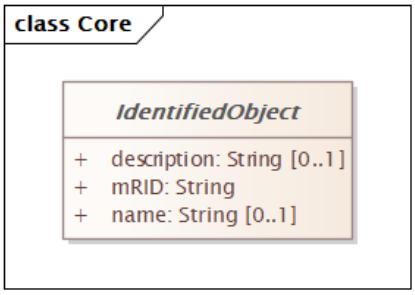
This package contains the availability schedule profile.



284

285 **Figure 1 – Class diagram AvailabilityScheduleProfile::AvailabilityScheduleProfile**

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



287

288 **Figure 2 – Class diagram AvailabilityScheduleProfile::Core**

289 Figure 2: The diagram shows classes from Base CIM used in the availability plan profile.

290 **3.2 (NC) AvailabilityAction**

291 Inheritance path = AvailabilityPowerSystemFunction : IdentifiedObject
292 Availability action is availability power system function of type grid state alteration (e.g. tap
293 position action).
294 Table 1 shows all attributes of AvailabilityAction.

295 **Table 1 – Attributes of AvailabilityScheduleProfile::AvailabilityAction**

name	mult	type	description
kind	1..1	AvailabilityFunctionKind	(NC) inherited from: AvailabilityPowerSystemFunction

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

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

Table 2 – Association ends of AvailabilityScheduleProfile::AvailabilityAction with other classes

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

3.3 (NC) AvailabilityEquipment

Inheritance path = AvailabilityPowerSystemFunction : IdentifiedObject

Availability equipment is availability power system function of type equipment (e.g. ACLineSegment).

Table 3 shows all attributes of AvailabilityEquipment.

Table 3 – 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 4 shows all association ends of AvailabilityEquipment with other classes.

Table 4 – 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

3.4 (NC) AvailabilityExceptionalLimit

Inheritance path = AvailabilityPowerSystemFunction : IdentifiedObject

Availability exceptional limit is availability power system function of type operational limit restriction (e.g. current limit on ACLineSegment terminal). This is limits that occur due to special

operational condition that is outside the normal dynamic rating. The exceptional limit could also be the cause of asset related issues, e.g. de-rating due to fault.
Table 5 shows all attributes of AvailabilityExceptionalLimit.

Table 5 – Attributes of AvailabilityScheduleProfile::AvailabilityExceptionalLimit

name	mult	type	description
value	1..1	Float	(NC) Value for the refered 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 6 shows all association ends of AvailabilityExceptionalLimit with other classes.

Table 6 – 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

3.5 (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 7 shows all attributes of AvailabilityGroup.

Table 7 – 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

3.6 (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 8 shows all attributes of AvailabilityPowerSystemFunction.

Table 8 – 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 9 shows all association ends of AvailabilityPowerSystemFunction with other classes.

Table 9 – Association ends of AvailabilityScheduleProfile::AvailabilityPowerSystemFunction with other classes

mult from	name	mult to	type	description
0..*	AvailabilityGroup	0..1	AvailabilityGroup	(NC) Availability group that link all related power system function controlled by all relevant system operators.
1..*	AvailabilitySchedule	1..1	AvailabilitySchedule	(NC) Availability schedule for this availability power system function.

3.7 (abstract,NC) AvailabilityRemedialAction root class

Availability remedial action is a remedial action that cancel or reschedule an availability schedule.

3.8 (NC) AvailabilityRemedialActionScheme

Inheritance path = AvailabilityPowerSystemFunction : IdentifiedObject
Availability remedial action scheme is availability power system function of type remedial action scheme (e.g. System Integrity Protection Scheme (SIPS)) For instance SIPS that is taken our-of-service due to communication issues.
Table 10 shows all attributes of AvailabilityRemedialActionScheme.

Table 10 – 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 11 shows all association ends of AvailabilityRemedialActionScheme with other classes.

Table 11 – 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

3.9 (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 12 shows all attributes of AvailabilitySchedule.

Table 12 – 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.
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.
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.
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 13 shows all association ends of AvailabilitySchedule with other classes.

Table 13 – 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 my require another operator to request there (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.
0..1	PlannedSchedule	0..1	EventSchedule	(NC) Planned schedule that relates to this planned availability schedule.
0..*	PrioritySchedule	0..1	AvailabilitySchedule	(NC) Priority schedule. This is the schedule that has the highest priority and the only valid if not cancelled.

3.10 (NC) AvailabilityContainer

Inheritance path = AvailabilityPowerSystemFunction : IdentifiedObject

Availability container is availability power system function of type equipment container (e.g. Line, Bay etc). This is used when multiple equipment in the relevant container are having the same availability.

Table 14 shows all attributes of AvailabilityContainer.

Table 14 – 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 15 shows all association ends of AvailabilityContainer with other classes.

Table 15 – 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

3.11 (abstract,NC) BaseIrregularTimeSeries

Inheritance path = BaseTimeSeries : IdentifiedObject

Time series that has irregular points in time.

Table 16 shows all attributes of BaseIrregularTimeSeries.

Table 16 – Attributes of AvailabilityScheduleProfile::BaseIrregularTimeSeries

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	inherited from: BaseTimeSeries
kind	1..1	BaseTimeSeriesKind	inherited from: BaseTimeSeries
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

3.12 (abstract,NC) BaseTimeSeries

Inheritance path = IdentifiedObject
Time series of values at points in time.
Table 17 shows all attributes of BaseTimeSeries.

Table 17 – Attributes of AvailabilityScheduleProfile::BaseTimeSeries

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	Kind of interpolation done between time point.
kind	1..1	BaseTimeSeriesKind	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

3.13 (abstract) Equipment root class

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

3.14 (abstract) EquipmentContainer root class

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

3.15 (NC) EventSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
Time series represent irregular event described by event points in time.
Table 18 shows all attributes of EventSchedule.

Table 18 – Attributes of AvailabilityScheduleProfile::EventSchedule

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	inherited from: BaseTimeSeries
kind	1..1	BaseTimeSeriesKind	inherited from: BaseTimeSeries
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

3.16 (NC) EventTimePoint root class

Event valid for a given point in time.
Table 19 shows all attributes of EventTimePoint.

Table 19 – 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 20 shows all association ends of EventTimePoint with other classes.

Table 20 – 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.

3.17 (abstract,NC) GridStateAlteration

Inheritance path = IdentifiedObject

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

Table 21 shows all attributes of GridStateAlteration.

Table 21 – Attributes of AvailabilityScheduleProfile::GridStateAlteration

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

3.18 (abstract,NC) GridStateAlterationCollection root class

A collection of grid state alterations.

3.19 (abstract) IdentifiedObject root class

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

Table 22 shows all attributes of IdentifiedObject.

Table 22 – 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. 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	mult	type	description
name	0..1	String	The name is any free human readable and possibly non unique text naming the object.

3.20 (abstract) OperationalLimit

Inheritance path = IdentifiedObject
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 23 shows all attributes of OperationalLimit.

Table 23 – Attributes of AvailabilityScheduleProfile::OperationalLimit

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

3.21 (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.

3.22 (NC) TimeSeriesInterpolationKind enumeration

Kinds of interpolation of values between two time point.
Table 24 shows all literals of TimeSeriesInterpolationKind.

Table 24 – Literals of AvailabilityScheduleProfile::TimeSeriesInterpolationKind

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

3.23 (NC) AvailabilityFunctionKind enumeration

Kind of availability that is affecting the function.
Table 25 shows all literals of AvailabilityFunctionKind.

Table 25 – 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.

3.24 (NC) AvailabilityScheduleCauseKind enumeration

The kinds of cause of the (un)availability schedule.
Table 26 shows all literals of AvailabilityScheduleCauseKind.

Table 26 – Literals of AvailabilityScheduleProfile::AvailabilityScheduleCauseKind

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

3.25 (NC) BaseTimeSeriesKind enumeration

Kind of time series.
Table 27 shows all literals of BaseTimeSeriesKind.

Table 27 – 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.

3.26 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 28 shows all literals of UnitMultiplier.

492 **Table 28 – Literals of AvailabilityScheduleProfile::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.

493

494 ## 3.27 UnitSymbol enumeration

495 The derived units defined for usage in the CIM. In some cases, the derived unit is equal to an
496 SI unit. Whenever possible, the standard derived symbol is used instead of the formula for the
497 derived unit. For example, the unit symbol Farad is defined as "F" instead of "CPerV". In cases
498 where a standard symbol does not exist for a derived unit, the formula for the unit is used as
499 the unit symbol. For example, density does not have a standard symbol and so it is represented
500 as "kgPerm3". With the exception of the "kg", which is an SI unit, the unit symbols do not contain
501 multipliers and therefore represent the base derived unit to which a multiplier can be applied as
502 a whole.
503 Every unit symbol is treated as an unparseable text as if it were a single-letter symbol. The
504 meaning of each unit symbol is defined by the accompanying descriptive text and not by the
505 text contents of the unit symbol.
506 To allow the widest possible range of serializations without requiring special character handling,
507 several substitutions are made which deviate from the format described in IEC 80000-1. The
508 division symbol "/" is replaced by the letters "Per". Exponents are written in plain text after the
509 unit as "m3" instead of being formatted as "m" with a superscript of 3 or introducing a symbol
510 as in "m^3". The degree symbol "°" is replaced with the letters "deg". Any clarification of the
511 meaning for a substitution is included in the description for the unit symbol.
512 Non-SI units are included in list of unit symbols to allow sources of data to be correctly labelled
513 with their non-SI units (for example, a GPS sensor that is reporting numbers that represent feet
514 instead of meters). This allows software to use the unit symbol information correctly convert
515 and scale the raw data of those sources into SI-based units.
516 The integer values are used for harmonization with IEC 61850.
517 Table 29 shows all literals of UnitSymbol.

518 **Table 29 – Literals of AvailabilityScheduleProfile::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).

literal	value	description
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).

literal	value	description
V2	67	Volt squared (W^2/A^2).
As	68	Ampere seconds (A·s).
A2	69	Amperes squared (A^2).
A2s	70	Ampere squared time in square amperes (A^2s).
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 ($kg·m^2$) (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’.

literal	value	description
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.

literal	value	description
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).

519

3.28 Seconds datatype

Time, in seconds.
Table 30 shows all attributes of Seconds.

Table 30 – 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)

3.29 Boolean primitive

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

3.30 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".

3.31 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.

3.32 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.

3.33 Integer primitive

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

3.34 Float primitive

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

3.35 String primitive

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

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

Intentionally left blank. Sample data will be updated at later stage.