## **IEEE 2030.5 Package**

The IEEE 2030.5 model is organized into function sets, represented by sub-packages. However, all structures are defined inside a single namespace.

Copyright

The IEEE Standards publication(s) ("Document") is approved by the IEEE Standards Association ("IEEE-SA") Standards Board and is published in accordance with established IEEE-SA Standards Board bylaws and operations procedures.

IEEE owns the copyright to this Document in all forms of media. Copyright in the text retrieved, displayed or output from this Document is owned by IEEE and is protected by the copyright laws of the United States and by international treaties. IEEE reserves all rights not expressly granted.

IEEE is providing the Document to you at no charge. However, the Document is not to be considered within the "Public Domain", as IEEE is, and at all times shall remain, the sole copyright holder in the Document.

Terms of Use

You may retrieve, download and print one (1) copy of each Document in this Program for your personal use. You may retain one (1) additional copy of this Document as your personal archive copy.

Except as allowed by the copyright laws of the United States of America or applicable international treaties, or as explicitly allowed in these Terms of Use, you may not further copy, prepare, and/or distribute copies of the Document, nor significant portions of the Document, in any form, without prior written permission from IEEE.

Requests for permission to reprint this Document, in whole or in part, or requests for a license to reproduce and/or distribute this Document, in any form, must be submitted via email to the Standards Licensing and Contracts (http://standards.ieee.org/contact/form.html), or in writing to:

IEEE-SA Licensing and Contracts

445 Hoes Lane

Piscataway, NJ 08855



**Figure B.1**—Version Information

### **DeviceCapability Package**

Contains definition of the objects used to convey the resources that are implemented by the publishing host.



**Figure B.Diagram Sequence**—DeviceCapability

**DeviceCapability Object** (FunctionSetAssignmentsBase)

Returned by the URI provided by DNS-SD, to allow clients to find the URIs to the resources in which they are interested.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

### **Common Package**

This package contains objects that are used in multiple function sets.

#### **Identification Package**

Contains super-classes that define the attributes common to categories of objects.



**Figure B.Diagram Sequence**—Identification

**IdentifiedObject Object** (Resource)

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

***description attribute*** *(String32) [0..1]*

The description is a human readable text describing or naming the object.

***mRID attribute*** *(mRIDType)*

The global identifier of the object.

***version attribute*** *(VersionType) [0..1]*

Contains the version number of the object. See the type definition for details.

**Link Object** ()

Links provide a reference, via URI, to another resource.

***href attribute*** *(anyURI) «XSDattribute»*

A URI reference.

**List Object** (Resource)

Container to hold a collection of object instances or references. See Design Pattern section for additional details.

***all attribute*** *(UInt32) «XSDattribute»*

The number specifying "all" of the items in the list. Required on a response to a GET, ignored otherwise.

***results attribute*** *(UInt32) «XSDattribute»*

Indicates the number of items in this page of results.

**ListLink Object** (Link)

ListLinks provide a reference, via URI, to a List.

***all attribute*** *(UInt32) [0..1] «XSDattribute»*

Indicates the total number of items in the referenced list. This attribute SHALL be present if the href is a local or relative URI. This attribute SHOULD NOT be present if the href is a remote or absolute URI, as the server may be unaware of changes to the value.

**Resource Object** ()

A resource is an addressable unit of information, either a collection (List) or instance of an object (identifiedObject, or simply, Resource)

***href attribute*** *(anyURI) [0..1] «XSDattribute»*

A reference to the resource address (URI). Required in a response to a GET, ignored otherwise.

**RespondableIdentifiedObject Object** (RespondableResource)

An IdentifiedObject to which a Response can be requested.

***description attribute*** *(String32) [0..1]*

The description is a human readable text describing or naming the object.

***mRID attribute*** *(mRIDType)*

The global identifier of the object.

***version attribute*** *(VersionType) [0..1]*

Contains the version number of the object. See the type definition for details.

**RespondableResource Object** (Resource)

A Resource to which a Response can be requested.

***replyTo attribute*** *(anyURI) [0..1] «XSDattribute»*

A reference to the response resource address (URI). Required on a response to a GET if responseRequired is "true".

***responseRequired attribute*** *(HexBinary8) [0..1] «XSDattribute»*

Indicates whether or not a response is required upon receipt, creation or update of this resource. Responses shall be posted to the collection specified in "replyTo".

If the resource has a deviceCategory field, devices that match one or more of the device types indicated in deviceCategory SHALL respond according to the rules listed below. If the category does not match, the device SHALL NOT respond. If the resource does not have a deviceCategory field, a device receiving the resource SHALL respond according to the rules listed below.

Value encoded as hex according to the following bit assignments, any combination is possible.

See Table 27 for the list of appropriate Response status codes to be sent for these purposes.

0 - End device shall indicate that message was received

1 - End device shall indicate specific response.

2 - End user / customer response is required.

All other values reserved.

**RespondableSubscribableIdentifiedObject Object** (RespondableResource)

An IdentifiedObject to which a Response can be requested.

***description attribute*** *(String32) [0..1]*

The description is a human readable text describing or naming the object.

***mRID attribute*** *(mRIDType)*

The global identifier of the object.

***subscribable attribute*** *(SubscribableType) [0..1] «XSDattribute»*

Indicates whether or not subscriptions are supported for this resource, and whether or not conditional (thresholds) are supported. If not specified, is "not subscribable" (0).

***version attribute*** *(VersionType) [0..1]*

Contains the version number of the object. See the type definition for details.

**SubscribableIdentifiedObject Object** (SubscribableResource)

An IdentifiedObject to which a Subscription can be requested.

***description attribute*** *(String32) [0..1]*

The description is a human readable text describing or naming the object.

***mRID attribute*** *(mRIDType)*

The global identifier of the object.

***version attribute*** *(VersionType) [0..1]*

Contains the version number of the object. See the type definition for details.

**SubscribableList Object** (SubscribableResource)

A List to which a Subscription can be requested.

***all attribute*** *(UInt32) «XSDattribute»*

The number specifying "all" of the items in the list. Required on GET, ignored otherwise.

***results attribute*** *(UInt32) «XSDattribute»*

Indicates the number of items in this page of results.

**SubscribableResource Object** (Resource)

A Resource to which a Subscription can be requested.

***subscribable attribute*** *(SubscribableType) [0..1] «XSDattribute»*

Indicates whether or not subscriptions are supported for this resource, and whether or not conditional (thresholds) are supported. If not specified, is "not subscribable" (0).

#### **Objects Package**

Contains definitions of objects used by multiple function sets.



**Figure B.Diagram Sequence**—Events



**Figure B.Diagram Sequence**—Programs



**Figure B.Diagram Sequence**—Error

**Error Object** ()

Contains information about the nature of an error if a request could not be completed successfully.

***maxRetryDuration attribute*** *(UInt16) [0..1]*

Contains the number of seconds the client SHOULD wait before retrying the request.

***reasonCode attribute*** *(UInt16)*

Code indicating the reason for failure.

0 - Invalid request format

1 - Invalid request values (e.g. invalid threshold values)

2 - Resource limit reached

3 - Conditional subscription field not supported

4 - Maximum request frequency exceeded

All other values reserved

**Event Object** (RespondableSubscribableIdentifiedObject)

An Event indicates information that applies to a particular period of time. Events SHALL be executed relative to the time of the server, as described in the Time function set section 11.1.

***creationTime attribute*** *(TimeType)*

The time at which the Event was created.

***interval attribute*** *(DateTimeInterval)*

The period during which the Event applies.

**EventStatus Object** ()

Current status information relevant to a specific object. The Status object is used to indicate the current status of an Event. Devices can read the containing resource (e.g. TextMessage) to get the most up to date status of the event. Devices can also subscribe to a specific resource instance to get updates when any of its attributes change, including the Status object.

***currentStatus attribute*** *(UInt8)*

Field representing the current status type.

0 = Scheduled

This status indicates that the event has been scheduled and the event has not yet started. The server SHALL set the event to this status when the event is first scheduled and persist until the event has become active or has been cancelled. For events with a start time less than or equal to the current time, this status SHALL never be indicated, the event SHALL start with a status of “Active”.

1 = Active

This status indicates that the event is currently active. The server SHALL set the event to this status when the event reaches its earliest Effective Start Time.

2 = Cancelled

When events are cancelled, the Status.dateTime attribute SHALL be set to the time the cancellation occurred, which cannot be in the future. The server is responsible for maintaining the cancelled event in its collection for the duration of the original event, or until the server has run out of space and needs to store a new event. Client devices SHALL be aware of Cancelled events, determine if the Cancelled event applies to them, and cancel the event immediately if applicable.

3 = Cancelled with Randomization

The server is responsible for maintaining the cancelled event in its collection for the duration of the Effective Scheduled Period. Client devices SHALL be aware of Cancelled with Randomization events, determine if the Cancelled event applies to them, and cancel the event immediately, using the larger of (absolute value of randomizeStart) and (absolute value of randomizeDuration) as the end randomization, in seconds. This Status.type SHALL NOT be used with "regular" Events, only with specializations of RandomizableEvent.

4 = Superseded

Events marked as Superseded by servers are events that may have been replaced by new events from the same program that target the exact same set of deviceCategory's (if applicable) AND DERControl controls (e.g., opModTargetW) (if applicable) and overlap for a given period of time. Servers SHALL mark an event as Superseded at the earliest Effective Start Time of the overlapping event. Servers are responsible for maintaining the Superseded event in their collection for the duration of the Effective Scheduled Period.

Client devices encountering a Superseded event SHALL terminate execution of the event immediately and commence execution of the new event immediately, unless the current time is within the start randomization window of the superseded event, in which case the client SHALL obey the start randomization of the new event. This Status.type SHALL NOT be used with TextMessage, since multiple text messages can be active.

All other values reserved.

***dateTime attribute*** *(TimeType)*

The dateTime attribute will provide a timestamp of when the current status was defined. dateTime MUST be set to the time at which the status change occurred, not a time in the future or past.

***potentiallySuperseded attribute*** *(boolean)*

Set to true by a server of this event if there are events that overlap this event in time and also overlap in some, but not all, deviceCategory's (if applicable) AND DERControl controls (e.g., opModTargetW) (if applicable) in the same function set instance.

***potentiallySupersededTime attribute*** *(TimeType) [0..1]*

Indicates the time that the potentiallySuperseded flag was set.

***reason attribute*** *(String192) [0..1]*

The Reason attribute allows a Service provider to provide a textual explanation of the status.

**RandomizableEvent Object** (Event)

An Event that can indicate time ranges over which the start time and duration SHALL be randomized.

***randomizeDuration attribute*** *(OneHourRangeType) [0..1]*

Number of seconds boundary inside which a random value must be selected to be applied to the associated interval duration, to avoid sudden synchronized demand changes. If related to price level changes, sign may be ignored. Valid range is -3600 to 3600. If not specified, 0 is the default.

***randomizeStart attribute*** *(OneHourRangeType) [0..1]*

Number of seconds boundary inside which a random value must be selected to be applied to the associated interval start time, to avoid sudden synchronized demand changes. If related to price level changes, sign may be ignored. Valid range is -3600 to 3600. If not specified, 0 is the default.

#### **Types Package**

Contains definitions of reusable data types.



**Figure B.Diagram Sequence**—Types

**AccumulationBehaviourType Object** (UInt8)

0 = Not Applicable (default, if not specified)

3 = Cumulative

The sum of the previous billing period values. Note: “Cumulative” is commonly used in conjunction with “demand.” Each demand reset causes the maximum demand value for the present billing period (since the last demand reset) to accumulate as an accumulative total of all maximum demands. So instead of “zeroing” the demand register, a demand reset has the affect of adding the present maximum demand to this accumulating total.

4 = DeltaData

The difference between the value at the end of the prescribed interval and the beginning of the interval. This is used for incremental interval data.

Note: One common application would be for load profile data, another use might be to report the number of events within an interval (such as the number of equipment energizations within the specified period of time.)

6 = Indicating

As if a needle is swung out on the meter face to a value to indicate the current value. (Note: An “indicating” value is typically measured over hundreds of milliseconds or greater, or may imply a “pusher” mechanism to capture a value. Compare this to “instantaneous” which is measured over a shorter period of time.)

9 = Summation

A form of accumulation which is selective with respect to time.

Note : “Summation” could be considered a specialization of “Bulk Quantity” according to the rules of inheritance where “Summation” selectively accumulates pulses over a timing pattern, and “BulkQuantity” accumulates pulses all of the time.

12 = Instantaneous

Typically measured over the fastest period of time allowed by the definition of the metric (usually milliseconds or tens of milliseconds.) (Note: “Instantaneous” was moved to attribute #3 in 61968-9Ed2 from attribute #1 in 61968-9Ed1.)

All other values reserved.

**ApplianceLoadReductionType Object** (UInt8)

0 - Delay Appliance Load

Parameter requesting the appliance to respond by providing a moderate load reduction for the duration of a delay period. Typically referring to a “non-emergency” event in which appliances can continue operating if already in a load consuming period.

1 - Temporary Appliance Load Reduction

Parameter requesting the appliance to respond by providing an aggressive load reduction for a short time period. Typically referring to an “emergency/spinning reserve” event in which an appliance should start shedding load if currently in a load consuming period.

\* Full definition of how appliances react when receiving each parameter is document in the EPA document - ENERGY STAR® Program Requirements, Product Specification for Residential Refrigerators and Freezers, Eligibility Criteria 5, Draft 2 Version 5.0.

All other values reserved.

**CommodityType Object** (UInt8)

0 = Not Applicable (default, if not specified)

1 = Electricity secondary metered value (a premises meter is typically on the low voltage, or secondary, side of a service transformer)

2 = Electricity primary metered value (measured on the high voltage, or primary, side of the service transformer)

4 = Air

7 = NaturalGas

8 = Propane

9 = PotableWater

10 = Steam

11 = WasteWater

12 = HeatingFluid

13 = CoolingFluid

All other values reserved.

**ConsumptionBlockType Object** (UInt8)

0 = Not Applicable (default, if not specified)

1 = Block 1

2 = Block 2

3 = Block 3

4 = Block 4

5 = Block 5

6 = Block 6

7 = Block 7

8 = Block 8

9 = Block 9

10 = Block 10

11 = Block 11

12 = Block 12

13 = Block 13

14 = Block 14

15 = Block 15

16 = Block 16

All other values reserved.

**CurrencyCode Object** (UInt16)

Follows codes defined in [ISO 4217].

0 - Not Applicable (default, if not specified)

36 - Australian Dollar

124 - Canadian Dollar

840 - US Dollar

978 - Euro

This is not a complete list.

**DataQualifierType Object** (UInt8)

0 = Not Applicable (default, if not specified)

2 = Average

8 = Maximum

9 = Minimum

12 = Normal

29 = Standard Deviation of a Population (typically indicated by a lower case sigma)

30 = Standard Deviation of a Sample Drawn from a Population (typically indicated by a lower case 's')

All other values reserved.

**DateTimeInterval Object** «Compound»()

Interval of date and time.

***duration attribute*** *(UInt32)*

Duration of the interval, in seconds.

***start attribute*** *(TimeType)*

Date and time of the start of the interval.

**DeviceCategoryType Object** (HexBinary32)

The Device category types defined.

Bit positions SHALL be defined as follows:

0 - Programmable Communicating Thermostat

1 - Strip Heaters

2 - Baseboard Heaters

3 - Water Heater

4 - Pool Pump

5 - Sauna

6 - Hot Tub

7 - Smart Appliance

8 - Irrigation Pump

9 - Managed Commercial and Industrial (C&I) Loads

10 - Simple Misc. (Residential On/Off) Loads

11 - Exterior Lighting

12 - Interior Lighting

13 - Load Control Switch

14 - Energy Management System

15 - Smart Energy Module

16 - Electric Vehicle

17 - EVSE

18 - Virtual or Mixed DER

19 - Reciprocating Engine

20 - Fuel Cell

21 - Photovoltaic System

22 - Combined Heat and Power

23 - Combined PV and Storage

24 - Other Generation System

25 - Other Storage System

All other values reserved.

**DstRuleType Object** (HexBinary32)

Bit map encoded rule from which is calculated the start or end time, within the current year, to which daylight savings time offset must be applied.

The rule encoding:

Bits 0 - 11: seconds 0 - 3599

Bits 12 - 16: hours 0 - 23

Bits 17 - 19: day of the week 0 = not applicable, 1 - 7 (Monday = 1)

Bits 20 - 24: day of the month 0 = not applicable, 1 - 31

Bits 25 - 27: operator (detailed below)

Bits 28 - 31: month 1 - 12

Rule value of 0xFFFFFFFF means rule processing/DST correction is disabled.

The operators:

0: DST starts/ends on the Day of the Month

1: DST starts/ends on the Day of the Week that is on or after the Day of the Month

2: DST starts/ends on the first occurrence of the Day of the Week in a month

3: DST starts/ends on the second occurrence of the Day of the Week in a month

4: DST starts/ends on the third occurrence of the Day of the Week in a month

5: DST starts/ends on the forth occurrence of the Day of the Week in a month

6: DST starts/ends on the fifth occurrence of the Day of the Week in a month

7: DST starts/ends on the last occurrence of the Day of the Week in a month

An example: DST starts on third Friday in March at 1:45 AM. The rule...

Seconds: 2700

Hours: 1

Day of Week: 5

Day of Month: 0

Operator: 4

Month: 3

**FlowDirectionType Object** (UInt8)

0 = Not Applicable (default, if not specified)

1 = Forward (delivered to customer)

19 = Reverse (received from customer)

All other values reserved.

**GPSLocationType Object** «Compound»()

Specifies a GPS location, expressed in WGS 84 coordinates.

***lat attribute*** *(String32)*

Specifies the latitude from equator. -90 (south) to +90 (north) in decimal degrees.

***lon attribute*** *(String32)*

Specifies the longitude from Greenwich Meridian. -180 (west) to +180 (east) in decimal degrees.

**KindType Object** (UInt8)

0 = Not Applicable (default, if not specified)

3 = Currency

8 = Demand

12 = Energy

37 = Power

All other values reserved.

**LocaleType Object** (String42)

[RFC 4646] identifier of a language-region

**mRIDType Object** (HexBinary128)

A master resource identifier. The IANA PEN [PEN] provider ID SHALL be specified in bits 0-31, the least-significant bits, and objects created by that provider SHALL be assigned unique IDs with the remaining 96 bits.

0xFFFFFFFFFFFFFFFFFFFFFFFF[XXXXXXXX], where [XXXXXXXX] is the PEN, is reserved for a object that is being created (e.g., a ReadingSet for the current time that is still accumulating).

Except for this special reserved identifier, each modification of an object (resource) representation MUST have a different "version".

**OneHourRangeType Object** (Int16)

A signed time offset, typically applied to a Time value, expressed in seconds, with range -3600 to 3600.

**PENType Object** (UInt32)

IANA Private Enterprise Number [PEN].

**PerCent Object** (UInt16)

Used for percentages, specified in hundredths of a percent, 0 - 10000. (10000 = 100%)

**PhaseCode Object** (UInt8)

0 = Not Applicable (default, if not specified)

32 = Phase C (and S2)

33 = Phase CN (and S2N)

40 = Phase CA

64 = Phase B

65 = Phase BN

66 = Phase BC

128 = Phase A (and S1)

129 = Phase AN (and S1N)

132 = Phase AB

224 = Phase ABC

All other values reserved.

**PINType Object** (UInt32)

6 digit unsigned decimal integer (0 - 999999).

(Note that this only requires 20 bits, if it can be allocated.)

**PowerOfTenMultiplierType Object** (Int8)

-9 = nano=x10^-9

-6 = micro=x10^-6

-3 = milli=x10^-3

0 = none=x1 (default, if not specified)

1 = deca=x10

2 = hecto=x100

3 = kilo=x1000

6 = Mega=x10^6

9 = Giga=x10^9

This is not a complete list. Any integer between -9 and 9 SHALL be supported, indicating the power of ten multiplier for the units.

**PrimacyType Object** (UInt8)

Values possible for indication of "Primary" provider:

0: In home energy management system

1: Contracted premises service provider

2: Non-contractual service provider

3 - 64: Reserved

65 - 191: User-defined

192 - 255: Reserved

Lower numbers indicate higher priority.

**RealEnergy Object** ()

Real electrical energy

***multiplier attribute*** *(PowerOfTenMultiplierType)*

Multiplier for 'unit'.

***value attribute*** *(UInt48)*

Value of the energy in Watt-hours. (uom 72)

**RoleFlagsType Object** (HexBinary16)

Specifies the roles that apply to a usage point.

Bit 0 - isMirror - SHALL be set if the server is not the measurement device

Bit 1 - isPremisesAggregationPoint - SHALL be set if the UsagePoint is the point of delivery for a premises

Bit 2 - isPEV - SHALL be set if the usage applies to an electric vehicle

Bit 3 - isDER - SHALL be set if the usage applies to a distributed energy resource, capable of delivering power to the grid.

Bit 4 - isRevenueQuality - SHALL be set if usage was measured by a device certified as revenue quality

Bit 5 - isDC - SHALL be set if the usage point measures direct current

Bit 6 - isSubmeter - SHALL be set if the usage point is not a premises aggregation point

Bit 7-15 - Reserved

**ServiceKind Object** (UInt8)

Service kind

0 - electricity

1 - gas

2 - water

3 - time

4 - pressure

5 - heat

6 - cooling

All other values reserved.

**SFDIType Object** (UInt40)

Unsigned integer, max inclusive 687194767359, which is 2^36-1 (68719476735), with added check digit. See Section 8.3.2 for check digit calculation.

**SignedPerCent Object** (Int16)

Used for signed percentages, specified in hundredths of a percent, -10000 - 10000. (10000 = 100%)

**SignedRealEnergy Object** ()

Real electrical energy, signed.

***multiplier attribute*** *(PowerOfTenMultiplierType)*

Multiplier for 'unit'.

***value attribute*** *(Int48)*

Value of the energy in Watt-hours. (uom 72)

**SubscribableType Object** «XSDsimpleType»(UInt8)

The subscribable values.

0 - Resource does not support subscriptions

1 - Resource supports non-conditional subscriptions

2 - Resource supports conditional subscriptions

3 - Resource supports both conditional and non-conditional subscriptions

All other values reserved.

**TimeOffsetType Object** (Int32)

A signed time offset, typically applied to a Time value, expressed in seconds.

**TimeType Object** (Int64)

Time is a signed 64 bit value representing the number of seconds since 0 hours, 0 minutes, 0 seconds, on the 1st of January, 1970, in UTC, not counting leap seconds.

**TOUType Object** (UInt8)

0 = Not Applicable (default, if not specified)

1 = TOU A

2 = TOU B

3 = TOU C

4 = TOU D

5 = TOU E

6 = TOU F

7 = TOU G

8 = TOU H

9 = TOU I

10 = TOU J

11 = TOU K

12 = TOU L

13 = TOU M

14 = TOU N

15 = TOU O

All other values reserved.

**UnitType Object** (UInt8)

The unit types defined for end device control target reductions.

0 - kWh

1 - kW

2 - Watts

3 - Cubic Meters

4 - Cubic Feet

5 - US Gallons

6 - Imperial Gallons

7 - BTUs

8 - Liters

9 - kPA (gauge)

10 - kPA (absolute)

11 - Mega Joule

12 - Unitless

All other values reserved.

**UnitValueType Object** ()

Type for specification of a specific value, with units and power of ten multiplier.

***multiplier attribute*** *(PowerOfTenMultiplierType)*

Multiplier for 'unit'.

***unit attribute*** *(UomType)*

Unit in symbol

***value attribute*** *(Int32)*

Value in units specified

**UomType Object** (UInt8)

The following values are recommended values sourced from the unit of measure enumeration in IEC 61968-9 [61968]. Other values from the unit of measure enumeration in IEC 61968-9 [61968] MAY be used.

0 = Not Applicable (default, if not specified)

5 = A (Current in Amperes (RMS))

6 = Kelvin (Temperature)

23 = Degrees Celsius (Relative temperature)

29 = Voltage

31 = J (Energy joule)

33 = Hz (Frequency)

38 =W (Real power in Watts)

42 = m3 (Cubic Meter)

61 = VA (Apparent power)

63 = var (Reactive power)

65 = CosTheta (Displacement Power Factor)

67 = V² (Volts squared)

69 = A² (Amp squared)

71 = VAh (Apparent energy)

72 = Wh (Real energy in Watt-hours)

73 = varh (Reactive energy)

106 = Ah (Ampere-hours / Available Charge)

119 = ft3 (Cubic Feet)

122 = ft3/h (Cubic Feet per Hour)

125 = m3/h (Cubic Meter per Hour)

128 = US gl (US Gallons)

129 = US gl/h (US Gallons per Hour)

130 = IMP gl (Imperial Gallons)

131 = IMP gl/h (Imperial Gallons per Hour)

132 = BTU

133 = BTU/h

134 = Liter

137 = L/h (Liters per Hour)

140 = PA(gauge)

155 = PA(absolute)

169 = Therm

**VersionType Object** (UInt16)

Version SHALL indicate a distinct identifier for each revision of an IdentifiedObject. If not specified, a default version of "0" (initial version) SHALL be assumed. Upon modification of any IdentifiedObject, the mRID SHALL remain the same, but the version SHALL be incremented. Servers MAY NOT modify objects that they did not create, unless they were notified of the change from the entity controlling the object's PEN.

#### **Primitive Types Package**

Contains definitions of primitive data types based on XML schema primitives.



**Figure B.Diagram Sequence**—Primitive Types

**HexBinary8 Object** «XSDsimpleType»(hexBinary)

An 8-bit field encoded as a hex string (2 hex characters). Where applicable, bit 0, or the least significant bit, goes on the right. Note that hexBinary requires pairs of hex characters, so an odd number of characters requires a leading "0".

**HexBinary16 Object** «XSDsimpleType»(hexBinary)

A 16-bit field encoded as a hex string (4 hex characters max). Where applicable, bit 0, or the least significant bit, goes on the right. Note that hexBinary requires pairs of hex characters, so an odd number of characters requires a leading "0".

**HexBinary32 Object** «XSDsimpleType»(hexBinary)

A 32-bit field encoded as a hex string (8 hex characters max). Where applicable, bit 0, or the least significant bit, goes on the right. Note that hexBinary requires pairs of hex characters, so an odd number of characters requires a leading "0".

**HexBinary48 Object** «XSDsimpleType»(hexBinary)

A 48-bit field encoded as a hex string (12 hex characters max). Where applicable, bit 0, or the least significant bit, goes on the right. Note that hexBinary requires pairs of hex characters, so an odd number of characters requires a leading "0".

**HexBinary64 Object** «XSDsimpleType»(hexBinary)

A 64-bit field encoded as a hex string (16 hex characters max). Where applicable, bit 0, or the least significant bit, goes on the right. Note that hexBinary requires pairs of hex characters, so an odd number of characters requires a leading "0".

**HexBinary128 Object** «XSDsimpleType»(hexBinary)

A 128-bit field encoded as a hex string (32 hex characters max). Where applicable, bit 0, or the least significant bit, goes on the right. Note that hexBinary requires pairs of hex characters, so an odd number of characters requires a leading "0".

**HexBinary160 Object** «XSDsimpleType»(hexBinary)

A 160-bit field encoded as a hex string (40 hex characters max). Where applicable, bit 0, or the least significant bit, goes on the right. Note that hexBinary requires pairs of hex characters, so an odd number of characters requires a leading "0".

**String6 Object** «XSDsimpleType»(string)

Character string of max length 6. In order to limit internal storage, implementations SHALL reduce the length of strings using multi-byte characters so that the string may be stored using "maxLength" octets in the given encoding.

**String16 Object** «XSDsimpleType»(string)

Character string of max length 16. In order to limit internal storage, implementations SHALL reduce the length of strings using multi-byte characters so that the string may be stored using "maxLength" octets in the given encoding.

**String20 Object** «XSDsimpleType»(string)

Character string of max length 20. In order to limit internal storage, implementations SHALL reduce the length of strings using multi-byte characters so that the string may be stored using "maxLength" octets in the given encoding.

**String32 Object** «XSDsimpleType»(string)

Character string of max length 32. In order to limit internal storage, implementations SHALL reduce the length of strings using multi-byte characters so that the string may be stored using "maxLength" octets in the given encoding.

**String42 Object** «XSDsimpleType»(string)

Character string of max length 42. In order to limit internal storage, implementations SHALL reduce the length of strings using multi-byte characters so that the string may be stored using "maxLength" octets in the given encoding.

**String192 Object** «XSDsimpleType»(string)

Character string of max length 192. For all string types, in order to limit internal storage, implementations SHALL reduce the length of strings using multi-byte characters so that the string may be stored using "maxLength" octets in the given encoding.

**UInt8 Object** «XSDsimpleType»(unsignedByte)

Unsigned integer, max inclusive 255 (2^8-1)

**UInt16 Object** «XSDsimpleType»(unsignedShort)

Unsigned integer, max inclusive 65535 (2^16-1)

**UInt32 Object** «XSDsimpleType»(unsignedInt)

Unsigned integer, max inclusive 4294967295 (2^32-1)

**UInt40 Object** «XSDsimpleType»(unsignedLong)

Unsigned integer, max inclusive 1099511627775 (2^40-1)

**UInt48 Object** «XSDsimpleType»(unsignedLong)

Unsigned integer, max inclusive 281474976710655 (2^48-1)

**UInt64 Object** «XSDsimpleType»(unsignedLong)

Unsigned integer, max inclusive 18446744073709551615 (2^64-1)

**Int8 Object** «XSDsimpleType»(byte)

Signed integer, min -128 max +127

**Int16 Object** «XSDsimpleType»(short)

Signed integer, min -32768 max +32767

**Int32 Object** «XSDsimpleType»(int)

Signed integer, max inclusive 2147483647 (2^31), min inclusive -2147483647 (same as xs:int)

**Int48 Object** «XSDsimpleType»(long)

Signed integer, max inclusive 140737488355328 (2^47), min inclusive -140737488355328

**Int64 Object** «XSDsimpleType»(long)

Signed integer, max inclusive 9223372036854775807 (2^63), min inclusive -9223372036854775808 (same as xs:long)

### **EndDevice Package**



**Figure B.Diagram Sequence**—SelfDevice



**Figure B.Diagram Sequence**—EndDevice

**AbstractDevice Object** (SubscribableResource)

The EndDevice providing the resources available within the DeviceCapabilities.

***deviceCategory attribute*** *(DeviceCategoryType) [0..1]*

This field is for use in devices that can adjust energy usage (e.g., demand response, distributed energy resources). For devices that do not respond to EndDeviceControls or DERControls (for instance, an ESI), this field should not have any bits set.

***lFDI attribute*** *(HexBinary160) [0..1]*

Long form of device identifier. See the Security section for additional details.

***sFDI attribute*** *(SFDIType)*

Short form of device identifier, WITH the checksum digit. See the Security section for additional details.

**DeviceStatus Object** (Resource)

Status of device

***changedTime attribute*** *(TimeType)*

The time at which the reported values were recorded.

***onCount attribute*** *(UInt16) [0..1]*

The number of times that the device has been turned on: Count of "device on" times, since the last time the counter was reset

***opState attribute*** *(UInt8) [0..1]*

Device operational state:

0 - Not applicable / Unknown

1 - Not operating

2 - Operating

3 - Starting up

4 - Shutting down

5 - At disconnect level

6 - kW ramping

7 - kVar ramping

***opTime attribute*** *(UInt32) [0..1]*

Total time device has operated: re-settable: Accumulated time in seconds since the last time the counter was reset.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

**EndDevice Object** (AbstractDevice)

Asset container that performs one or more end device functions. Contains information about individual devices in the network.

***changedTime attribute*** *(TimeType)*

The time at which this resource was last modified or created.

***enabled attribute*** *(boolean) [0..1]*

This attribute indicates whether or not an EndDevice is enabled, or registered, on the server. If a server sets this attribute to false, the device is no longer registered. It should be noted that servers can delete EndDevice instances, but using this attribute for some time is more convenient for clients.

***postRate attribute*** *(UInt32) [0..1]*

POST rate, or how often EndDevice and subordinate resources should be POSTed, in seconds. A client MAY indicate a preferred postRate when POSTing EndDevice. A server MAY add or modify postRate to indicate its preferred posting rate.

**EndDeviceList Object** (SubscribableList)

A List element to hold EndDevice objects.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

**Registration Object** (Resource)

Registration represents an authorization to access the resources on a host.

***dateTimeRegistered attribute*** *(TimeType)*

Contains the time at which this registration was created, by which clients MAY prioritize information providers with the most recent registrations, when no additional direction from the consumer is available.

***pIN attribute*** *(PINType)*

Contains the registration PIN number associated with the device, including the checksum digit.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

**SelfDevice Object** (AbstractDevice)

The EndDevice providing the resources available within the DeviceCapabilities.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

**Temperature Object** ()

Specification of a temperature.

***multiplier attribute*** *(PowerOfTenMultiplierType)*

Multiplier for 'unit'.

***subject attribute*** *(UInt8)*

The subject of the temperature measurement

0 - Enclosure

1 - Transformer

2 - HeatSink

***value attribute*** *(Int16)*

Value in Degrees Celsius (uom 23).

### **FunctionSetAssignments Package**



**Figure B.Diagram Sequence**—FunctionSetAssignments

**FunctionSetAssignmentsBase Object** (Resource)

Defines a collection of function set instances that are to be used by one or more devices as indicated by the EndDevice object(s) of the server.

**FunctionSetAssignments Object** (FunctionSetAssignmentsBase)

Provides an identifiable, subscribable collection of resources for a particular device to consume.

***description attribute*** *(String32) [0..1]*

The description is a human readable text describing or naming the object.

***mRID attribute*** *(mRIDType)*

The global identifier of the object.

***subscribable attribute*** *(SubscribableType) [0..1] «XSDattribute»*

Indicates whether or not subscriptions are supported for this resource, and whether or not conditional (thresholds) are supported. If not specified, is "not subscribable" (0).

***version attribute*** *(VersionType) [0..1]*

Contains the version number of the object. See the type definition for details.

**FunctionSetAssignmentsList Object** (SubscribableList)

A List element to hold FunctionSetAssignments objects.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

### **Pub-Sub Package**

Contains resource definitions used to allow subscriptions and notifications of publications.



**Figure B.Diagram Sequence**—Pub-Sub

**Condition Object** ()

Indicates a condition that must be satisfied for the Notification to be triggered.

***attributeIdentifier attribute*** *(UInt8)*

0 = Reading value

1-255 = Reserved

***lowerThreshold attribute*** *(Int48)*

The value of the lower threshold

***upperThreshold attribute*** *(Int48)*

The value of the upper threshold

**SubscriptionBase Object** (Resource)

Holds the information related to a client subscription to receive updates to a resource automatically. The actual resources may be passed in the Notification by specifying a specific xsi:type for the Resource and passing the full representation.

***subscribedResource attribute*** *(anyURI)*

The resource for which the subscription applies. Query string parameters SHALL NOT be specified when subscribing to list resources. Should a query string parameter be specified, servers SHALL ignore them.

**Subscription Object** (SubscriptionBase)

Holds the information related to a client subscription to receive updates to a resource automatically.

***encoding attribute*** *(UInt8)*

0 - application/sep+xml

1 - application/sep-exi

2-255 - reserved

***level attribute*** *(String16)*

Contains the preferred schema and extensibility level indication such as "+S1"

***limit attribute*** *(UInt32)*

This element is used to indicate the maximum number of list items that should be included in a notification when the subscribed resource changes. This limit is meant to be functionally equivalent to the ‘limit’ query string parameter, but applies to both list resources as well as other resources. For list resources, if a limit of ‘0’ is specified, then notifications SHALL contain a list resource with results=’0’ (equivalent to a simple change notification). For list resources, if a limit greater than ‘0’ is specified, then notifications SHALL contain a list resource with results equal to the limit specified (or less, should the list contain fewer items than the limit specified or should the server be unable to provide the requested number of items for any reason) and follow the same rules for list resources (e.g., ordering). For non-list resources, if a limit of ‘0’ is specified, then notifications SHALL NOT contain a resource representation (equivalent to a simple change notification). For non-list resources, if a limit greater than ‘0’ is specified, then notifications SHALL contain the representation of the changed resource.

***notificationURI attribute*** *(anyURI)*

The resource to which to post the notifications about the requested subscribed resource. Because this URI will exist on a server other than the one being POSTed to, this attribute SHALL be a fully-qualified absolute URI, not a relative reference.

**SubscriptionList Object** (List)

A List element to hold Subscription objects.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

**Notification Object** (SubscriptionBase)

Holds the information related to a client subscription to receive updates to a resource automatically. The actual resources may be passed in the Notification by specifying a specific xsi:type for the Resource and passing the full representation.

***newResourceURI attribute*** *(anyURI) [0..1]*

The new location of the resource, if moved. This attribute SHALL be a fully-qualified absolute URI, not a relative reference.

***status attribute*** *(UInt8)*

0 = Default Status

1 = Subscription canceled, no additional information

2 = Subscription canceled, resource moved

3 = Subscription canceled, resource definition changed (e.g., a new version of IEEE 2030.5)

4 = Subscription canceled, resource deleted

All other values reserved.

***subscriptionURI attribute*** *(anyURI)*

The subscription from which this notification was triggered. This attribute SHALL be a fully-qualified absolute URI, not a relative reference.

**NotificationList Object** (List)

A List element to hold Notification objects.

### **Response Package**

Contains definitions of objects enabling responses to be sent back to suppliers and providers.



**Figure B.Diagram Sequence**—Response

**DERControlResponse Object** (Response)

A response to a DERControl

**FlowReservationResponseResponse Object** (Response)

A response to a FlowReservationResponse

**AppliedTargetReduction Object** ()

Specifies the value of the TargetReduction applied by the device.

***type attribute*** *(UnitType)*

Enumerated field representing the type of reduction requested.

***value attribute*** *(UInt16)*

Indicates the requested amount of the relevant commodity to be reduced.

**DrResponse Object** (Response)

A response to a Demand Response Load Control (EndDeviceControl) message.

***overrideDuration attribute*** *(UInt16) [0..1]*

Indicates the amount of time, in seconds, that the client partially opts-out during the demand response event. When overriding within the allowed override duration, the client SHALL send a partial opt-out (Response status code 8) for partial opt-out upon completion, with the total time the event was overridden (this attribute) populated. The client SHALL send a no participation status response (status type 10) if the user partially opts-out for longer than EndDeviceControl.overrideDuration.

**PriceResponse Object** (Response)

A response related to a price message.

**Response Object** (Resource)

The Response object is the generic response data repository which is extended for specific function sets.

***createdDateTime attribute*** *(TimeType) [0..1]*

The createdDateTime field contains the date and time when the acknowledgement/status occurred in the client. The client will provide the timestamp to ensure the proper time is captured in case the response is delayed in reaching the server (server receipt time would not be the same as the actual confirmation time). The time reported from the client should be relative to the time server indicated by the FunctionSetAssignment that also indicated the event resource; if no FunctionSetAssignment exists, the time of the server where the event resource was hosted.

***endDeviceLFDI attribute*** *(HexBinary160)*

Contains the LFDI of the device providing the response.

***status attribute*** *(UInt8) [0..1]*

The status field contains the acknowledgement or status. Each event type (DRLC, DER, Price, or Text) can return different status information (e.g. an Acknowledge will be returned for a Price event where a DRLC event can return Event Received, Event Started, and Event Completed). The Status field value definitions are defined in Table 27: Response Types by Function Set.

***subject attribute*** *(mRIDType)*

The subject field provides a method to match the response with the originating event. It is populated with the mRID of the original object.

**ResponseList Object** (List)

A List element to hold Response objects.

**ResponseSet Object** (IdentifiedObject)

A container for a ResponseList.

**ResponseSetList Object** (List)

A List element to hold ResponseSet objects.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

**TextResponse Object** (Response)

A response to a text message

### **Time Package**



**Figure B.Diagram Sequence**—Time

**Time Object** (Resource)

Contains the representation of time, constantly updated.

***currentTime attribute*** *(TimeType)*

The current time, in the format defined by TimeType.

***dstEndTime attribute*** *(TimeType)*

Time at which daylight savings ends (dstOffset no longer applied). Result of dstEndRule calculation.

***dstOffset attribute*** *(TimeOffsetType)*

Daylight savings time offset from local standard time. A typical practice is advancing clocks one hour when daylight savings time is in effect, which would result in a positive dstOffset.

***dstStartTime attribute*** *(TimeType)*

Time at which daylight savings begins (apply dstOffset). Result of dstStartRule calculation.

***localTime attribute*** *(TimeType) [0..1]*

Local time: localTime = currentTime + tzOffset (+ dstOffset when in effect).

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

***quality attribute*** *(UInt8)*

Metric indicating the quality of the time source from which the service acquired time. Lower (smaller) quality enumeration values are assumed to be more accurate.

3 - time obtained from external authoritative source such as NTP

4 - time obtained from level 3 source

5 - time manually set or obtained from level 4 source

6 - time obtained from level 5 source

7 - time intentionally uncoordinated

All other values are reserved for future use.

***tzOffset attribute*** *(TimeOffsetType)*

Local time zone offset from currentTime. Does not include any daylight savings time offsets. For American time zones, a negative tzOffset SHALL be used (eg, EST = GMT-5 which is -18000).

### **DeviceInformation Package**

Contains general information about devices.



**Figure B.Diagram Sequence**—DeviceInformation

**DeviceInformation Object** (Resource)

Contains identification and other information about the device that changes very infrequently, typically only when updates are applied, if ever.

***functionsImplemented attribute*** *(HexBinary64) [0..1]*

Bitmap indicating the function sets used by the device as a client.

0 - Device Capability

1 - Self Device Resource

2 - End Device Resource

3 - Function Set Assignments

4 - Subscription/Notification Mechanism

5 - Response

6 - Time

7 - Device Information

8 - Power Status

9 - Network Status

10 - Log Event

11 - Configuration Resource

12 - Software Download

13 - DRLC

14 - Metering

15 - Pricing

16 - Messaging

17 - Billing

18 - Prepayment

19 - Flow Reservation

20 - DER Control

***gpsLocation attribute*** *(GPSLocationType) [0..1]*

GPS location of this device.

***lFDI attribute*** *(HexBinary160)*

Long form device identifier. See the Security section for full details.

***mfDate attribute*** *(TimeType)*

Date/time of manufacture

***mfHwVer attribute*** *(String32)*

Manufacturer hardware version

***mfID attribute*** *(PENType)*

The manufacturer's IANA Enterprise Number.

***mfInfo attribute*** *(String32) [0..1]*

Manufacturer dependent information related to the manufacture of this device

***mfModel attribute*** *(String32)*

Manufacturer's model number

***mfSerNum attribute*** *(String32)*

Manufacturer assigned serial number

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

***primaryPower attribute*** *(PowerSourceType)*

Primary source of power.

***secondaryPower attribute*** *(PowerSourceType)*

Secondary source of power

***swActTime attribute*** *(TimeType)*

Activation date/time of currently running software

***swVer attribute*** *(String32)*

Currently running software version

**DRLCCapabilities Object** ()

Contains information about the static capabilities of the device, to allow service providers to know what types of functions are supported, what the normal operating ranges and limits are, and other similar information, in order to provide better suggestions of applicable programs to receive the maximum benefit.

***averageEnergy attribute*** *(RealEnergy)*

The average hourly energy usage when in normal operating mode.

***maxDemand attribute*** *(ActivePower)*

The maximum demand rating of this end device.

***optionsImplemented attribute*** *(HexBinary32)*

Bitmap indicating the DRLC options implemented by the device.

0 - Target reduction (kWh)

1 - Target reduction (kW)

2 - Target reduction (Watts)

3 - Target reduction (Cubic Meters)

4 - Target reduction (Cubic Feet)

5 - Target reduction (US Gallons)

6 - Target reduction (Imperial Gallons)

7 - Target reduction (BTUs)

8 - Target reduction (Liters)

9 - Target reduction (kPA (gauge))

10 - Target reduction (kPA (absolute))

11 - Target reduction (Mega Joule)

12 - Target reduction (Unitless)

13-15 - Reserved

16 - Temperature set point

17 - Temperature offset

18 - Duty cycle

19 - Load adjustment percentage

20 - Appliance load reduction

21-31 - Reserved

**SupportedLocale Object** (Resource)

Specifies a locale that is supported

***locale attribute*** *(LocaleType)*

The code for a locale that is supported

**SupportedLocaleList Object** (List)

A List element to hold SupportedLocale objects.

### **PowerStatus Package**



**Figure B.Diagram Sequence**—PowerStatus

**PowerStatus Object** (Resource)

Contains the status of the device's power sources

***batteryStatus attribute*** *(UInt8)*

Battery system status

0 = unknown

1 = normal (more than LowChargeThreshold remaining)

2 = low (less than LowChargeThreshold remaining)

3 = depleted (0% charge remaining)

4 = not applicable (mains powered only)

***changedTime attribute*** *(TimeType)*

The time at which the reported values were recorded.

***currentPowerSource attribute*** *(PowerSourceType)*

This value will be fixed for devices powered by a single source. This value may change for devices able to transition between multiple power sources (mains to battery backup, etc.).

***estimatedChargeRemaining attribute*** *(PerCent) [0..1]*

Estimate of remaining battery charge as a percent of full charge.

***estimatedTimeRemaining attribute*** *(UInt32) [0..1]*

Estimated time (in seconds) to total battery charge depletion (under current load)

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

***sessionTimeOnBattery attribute*** *(UInt32) [0..1]*

If the device has a battery, this is the time since the device last switched to battery power, or the time since the device was restarted, whichever is less, in seconds.

***totalTimeOnBattery attribute*** *(UInt32) [0..1]*

If the device has a battery, this is the total time the device has been on battery power, in seconds. It may be reset when the battery is replaced.

**PowerSourceType Object** (UInt8)

0 - none

1 - mains

2 - battery

3 - local generation

4 - emergency

5 - unknown

All other values reserved.

**PEVInfo Object** ()

Contains attributes that can be exposed by PEVs and other devices that have charging requirements.

***chargingPowerNow attribute*** *(ActivePower)*

This is the actual power flow in or out of the charger or inverter. This is calculated by the vehicle based on actual measurements. This number is positive for charging.

***energyRequestNow attribute*** *(RealEnergy)*

This is the amount of energy that must be transferred from the grid to EVSE and PEV to achieve the target state of charge allowing for charger efficiency and any vehicle and EVSE parasitic loads. This is calculated by the vehicle and changes throughout the connection as forward or reverse power flow change the battery state of charge. This number is positive for charging.

***maxForwardPower attribute*** *(ActivePower)*

This is maximum power transfer capability that could be used for charging the PEV to perform the requested energy transfer. It is the lower of the vehicle or EVSE physical power limitations. It is not based on economic considerations. The vehicle may draw less power than this value based on its charging cycle. The vehicle defines this parameter. This number is positive for charging power flow.

***minimumChargingDuration attribute*** *(UInt32)*

This is computed by the PEV based on the charging profile to complete the energy transfer if the maximum power is authorized. The value will never be smaller than the ratio of the energy request to the power request because the charging profile may not allow the maximum power to be used throughout the transfer. This is a critical parameter for determining whether any slack time exists in the charging cycle between the current time and the TCIN.

***targetStateOfCharge attribute*** *(PerCent)*

This is the target state of charge that is to be achieved during charging before the time of departure (TCIN). The default value is 100%. The value cannot be set to a value less than the actual state of charge.

***timeChargeIsNeeded attribute*** *(TimeType)*

Time Charge is Needed (TCIN) is the time that the PEV is expected to depart. The value is manually entered using controls and displays in the vehicle or on the EVSE or using a mobile device. It is authenticated and saved by the PEV. This value may be updated during a charging session.

***timeChargingStatusPEV attribute*** *(TimeType)*

This is the time that the parameters are updated, except for changes to TCIN.

### **NetworkStatus Package**



**Figure B.Diagram Sequence**—NetworkStatus

**IEEE\_802\_15\_4 Object** ()

Contains 802.15.4 link layer specific attributes.

***capabilityInfo attribute*** *(HexBinary8)*

As defined by IEEE 802.15.4

***shortAddress attribute*** *(UInt16)*

As defined by IEEE 802.15.4

**IPAddr Object** (Resource)

An Internet Protocol address object.

***address attribute*** *(HexBinary128)*

An IP address value.

**IPAddrList Object** (List)

List of IPAddr instances.

**IPInterface Object** (Resource)

Specific IPInterface resource. This resource may be thought of as network status information for a specific network (IP) layer interface.

***ifDescr attribute*** *(String192) [0..1]*

Use rules from [RFC 2863].

***ifHighSpeed attribute*** *(UInt32) [0..1]*

Use rules from [RFC 2863].

***ifInBroadcastPkts attribute*** *(UInt32) [0..1]*

Use rules from [RFC 2863].

***ifIndex attribute*** *(UInt32) [0..1]*

Use rules from [RFC 2863].

***ifInDiscards attribute*** *(UInt32) [0..1]*

Use rules from [RFC 2863]. Can be thought of as Input Datagrams Discarded.

***ifInErrors attribute*** *(UInt32) [0..1]*

Use rules from [RFC 2863].

***ifInMulticastPkts attribute*** *(UInt32) [0..1]*

Use rules from [RFC 2863]. Can be thought of as Multicast Datagrams Received.

***ifInOctets attribute*** *(UInt32) [0..1]*

Use rules from [RFC 2863]. Can be thought of as Bytes Received.

***ifInUcastPkts attribute*** *(UInt32) [0..1]*

Use rules from [RFC 2863]. Can be thought of as Datagrams Received.

***ifInUnknownProtos attribute*** *(UInt32) [0..1]*

Use rules from [RFC 2863]. Can be thought of as Datagrams with Unknown Protocol Received.

***ifMtu attribute*** *(UInt32) [0..1]*

Use rules from [RFC 2863].

***ifName attribute*** *(String16) [0..1]*

Use rules from [RFC 2863].

***ifOperStatus attribute*** *(UInt8) [0..1]*

Use rules and assignments from [RFC 2863].

***ifOutBroadcastPkts attribute*** *(UInt32) [0..1]*

Use rules from [RFC 2863]. Can be thought of as Broadcast Datagrams Sent.

***ifOutDiscards attribute*** *(UInt32) [0..1]*

Use rules from [RFC 2863]. Can be thought of as Output Datagrams Discarded.

***ifOutErrors attribute*** *(UInt32) [0..1]*

Use rules from [RFC 2863].

***ifOutMulticastPkts attribute*** *(UInt32) [0..1]*

Use rules from [RFC 2863]. Can be thought of as Multicast Datagrams Sent.

***ifOutOctets attribute*** *(UInt32) [0..1]*

Use rules from [RFC 2863]. Can be thought of as Bytes Sent.

***ifOutUcastPkts attribute*** *(UInt32) [0..1]*

Use rules from [RFC 2863]. Can be thought of as Datagrams Sent.

***ifPromiscuousMode attribute*** *(boolean) [0..1]*

Use rules from [RFC 2863].

***ifSpeed attribute*** *(UInt32) [0..1]*

Use rules from [RFC 2863].

***ifType attribute*** *(UInt16) [0..1]*

Use rules and assignments from [RFC 2863].

***lastResetTime attribute*** *(Int64) [0..1]*

Similar to ifLastChange in [RFC 2863].

***lastUpdatedTime attribute*** *(Int64) [0..1]*

The date/time of the reported status.

**IPInterfaceList Object** (List)

List of IPInterface instances.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

**LLInterface Object** (Resource)

A link-layer interface object.

***CRCerrors attribute*** *(UInt32)*

Contains the number of CRC errors since reset.

***EUI64 attribute*** *(HexBinary64)*

Contains the EUI-64 of the link layer interface. 48 bit MAC addresses SHALL be changed into an EUI-64 using the method defined in [RFC 4291], Appendix A. (The method is to insert "0xFFFE" as described in the reference.)

***linkLayerType attribute*** *(UInt8)*

Specifies the type of link layer interface associated with the IPInterface. Values are below.

0 = Unspecified

1 = IEEE 802.3 (Ethernet)

2 = IEEE 802.11 (WLAN)

3 = IEEE 802.15 (PAN)

4 = IEEE 1901 (PLC)

All other values reserved.

***LLAckNotRx attribute*** *(UInt32) [0..1]*

Number of times an ACK was not received for a frame transmitted (when ACK was requested).

***LLCSMAFail attribute*** *(UInt32) [0..1]*

Number of times CSMA failed.

***LLFramesDropRx attribute*** *(UInt32) [0..1]*

Number of dropped receive frames.

***LLFramesDropTx attribute*** *(UInt32) [0..1]*

Number of dropped transmit frames.

***LLFramesRx attribute*** *(UInt32) [0..1]*

Number of link layer frames received.

***LLFramesTx attribute*** *(UInt32) [0..1]*

Number of link layer frames transmitted.

***LLMediaAccessFail attribute*** *(UInt32) [0..1]*

Number of times access to media failed.

***LLOctetsRx attribute*** *(UInt32) [0..1]*

Number of Bytes received.

***LLOctetsTx attribute*** *(UInt32) [0..1]*

Number of Bytes transmitted.

***LLRetryCount attribute*** *(UInt32) [0..1]*

Number of MAC transmit retries.

***LLSecurityErrorRx attribute*** *(UInt32) [0..1]*

Number of receive security errors.

**LLInterfaceList Object** (List)

List of LLInterface instances.

**loWPAN Object** ()

Contains information specific to 6LoWPAN.

***octetsRx attribute*** *(UInt32) [0..1]*

Number of Bytes received

***octetsTx attribute*** *(UInt32) [0..1]*

Number of Bytes transmitted

***packetsRx attribute*** *(UInt32)*

Number of packets received

***packetsTx attribute*** *(UInt32)*

Number of packets transmitted

***rxFragError attribute*** *(UInt32)*

Number of errors receiving fragments

**Neighbor Object** (Resource)

Contains 802.15.4 link layer specific attributes.

***isChild attribute*** *(boolean)*

True if the neighbor is a child.

***linkQuality attribute*** *(UInt8)*

The quality of the link, as defined by 802.15.4

***shortAddress attribute*** *(UInt16)*

As defined by IEEE 802.15.4

**NeighborList Object** (List)

List of 15.4 neighbors.

**RPLInstance Object** (Resource)

Specific RPLInstance resource. This resource may be thought of as network status information for a specific RPL instance associated with IPInterface.

***DODAGid attribute*** *(UInt8)*

See [RFC 6550].

***DODAGroot attribute*** *(boolean)*

See [RFC 6550].

***flags attribute*** *(UInt8)*

See [RFC 6550].

***groundedFlag attribute*** *(boolean)*

See [RFC 6550].

***MOP attribute*** *(UInt8)*

See [RFC 6550].

***PRF attribute*** *(UInt8)*

See [RFC 6550].

***rank attribute*** *(UInt16)*

See [RFC 6550].

***RPLInstanceID attribute*** *(UInt8)*

See [RFC 6550].

***versionNumber attribute*** *(UInt8)*

See [RFC 6550].

**RPLInstanceList Object** (List)

List of RPLInstances associated with the IPinterface.

**RPLSourceRoutes Object** (Resource)

A RPL source routes object.

***DestAddress attribute*** *(HexBinary128)*

See [RFC 6554].

***SourceRoute attribute*** *(HexBinary128)*

See [RFC 6554].

**RPLSourceRoutesList Object** (List)

List or RPL source routes if the hosting device is the DODAGroot

### **LogEvents Package**



**Figure B.Diagram Sequence**—LogEvents

**LogEvent Object** (Resource)

A time stamped instance of a significant event detected by the device.

***createdDateTime attribute*** *(TimeType)*

The date and time that the event occurred.

***details attribute*** *(String32) [0..1]*

Human readable text that MAY be used to transmit additional details about the event. A host MAY remove this field when received.

***extendedData attribute*** *(UInt32) [0..1]*

May be used to transmit additional details about the event.

***functionSet attribute*** *(UInt8)*

If the profileID indicates this is IEEE 2030.5, the functionSet is defined by IEEE 2030.5 and SHALL be one of the values from the table below (IEEE 2030.5 function set identifiers). If the profileID is anything else, the functionSet is defined by the identified profile.

0 General (not specific to a function set)

1 Publish and Subscribe

2 End Device

3 Function Set Assignment

4 Response

5 Demand Response and Load Control

6 Metering

7 Pricing

8 Messaging

9 Billing

10 Prepayment

11 Distributed Energy Resources

12 Time

13 Software Download

14 Device Information

15 Power Status

16 Network Status

17 Log Event List

18 Configuration

19 Security

All other values are reserved.

***logEventCode attribute*** *(UInt8)*

An 8 bit unsigned integer. logEventCodes are scoped to a profile and a function set. If the profile is IEEE 2030.5, the logEventCode is defined by IEEE 2030.5 within one of the function sets of IEEE 2030.5. If the profile is anything else, the logEventCode is defined by the specified profile.

***logEventID attribute*** *(UInt16)*

This 16-bit value, combined with createdDateTime, profileID, and logEventPEN, should provide a reasonable level of uniqueness.

***logEventPEN attribute*** *(PENType)*

The Private Enterprise Number(PEN) of the entity that defined the profileID, functionSet, and logEventCode of the logEvent. IEEE 2030.5-assigned logEventCodes SHALL use the IEEE 2030.5 PEN. Combinations of profileID, functionSet, and logEventCode SHALL have unique meaning within a logEventPEN and are defined by the owner of the PEN.

***profileID attribute*** *(UInt8)*

The profileID identifies which profile (HA, BA, SE, etc) defines the following event information.

0 Not profile specific.

1 Vendor Defined

2 IEEE 2030.5

3 Home Automation

4 Building Automation

All other values are reserved.

**LogEventList Object** (SubscribableList)

A List element to hold LogEvent objects.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

### **Configuration Package**



**Figure B.Diagram Sequence**—Configuration

**Configuration Object** (SubscribableResource)

This resource contains various settings to control the operation of the device

***currentLocale attribute*** *(LocaleType)*

[RFC 4646] identifier of the language-region currently in use.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

***userDeviceName attribute*** *(String32)*

User assigned, convenience name used for network browsing displays, etc. Example "My Thermostat"

**PowerConfiguration Object** ()

Contains configuration related to the device's power sources

***batteryInstallTime attribute*** *(TimeType) [0..1]*

Time/Date at which battery was installed,

***lowChargeThreshold attribute*** *(UInt32) [0..1]*

In context of the PowerStatus resource, this is the value of EstimatedTimeRemaining below which BatteryStatus "low" is indicated and the PS\_LOW\_BATTERY is raised.

**PriceResponseCfg Object** (Resource)

Configuration data that specifies how price responsive devices SHOULD respond to price changes while acting upon a given RateComponent.

***consumeThreshold attribute*** *(Int32)*

Price responsive clients acting upon the associated RateComponent SHOULD consume the associated commodity while the price is less than this threshold.

***maxReductionThreshold attribute*** *(Int32)*

Price responsive clients acting upon the associated RateComponent SHOULD reduce consumption to the maximum extent possible while the price is greater than this threshold.

**PriceResponseCfgList Object** (List)

A List element to hold PriceResponseCfg objects.

**TimeConfiguration Object** ()

Contains attributes related to the configuration of the time service.

***dstEndRule attribute*** *(DstRuleType)*

Rule to calculate end of daylight savings time in the current year. Result of dstEndRule must be greater than result of dstStartRule.

***dstOffset attribute*** *(TimeOffsetType)*

Daylight savings time offset from local standard time.

***dstStartRule attribute*** *(DstRuleType)*

Rule to calculate start of daylight savings time in the current year. Result of dstEndRule must be greater than result of dstStartRule.

***tzOffset attribute*** *(TimeOffsetType)*

Local time zone offset from UTCTime. Does not include any daylight savings time offsets.

### **SoftwareDownload Package**



**Figure B.Diagram Sequence**—Files

**File Object** (Resource)

This resource contains various meta-data describing a file's characteristics. The meta-data provides general file information and also is used to support filtered queries of file lists

***activateTime attribute*** *(TimeType) [0..1]*

This element MUST be set to the date/time at which this file is activated. If the activation time is less than or equal to current time, the LD MUST immediately place the file into the activated state (in the case of a firmware file, the file is now the running image). If the activation time is greater than the current time, the LD MUST wait until the specified activation time is reached, then MUST place the file into the activated state. Omission of this element means that the LD MUST NOT take any action to activate the file until a subsequent GET to this File resource provides an activateTime.

***fileURI attribute*** *(anyURI)*

This element MUST be set to the URI location of the file binary artifact. This is the BLOB (binary large object) that is actually loaded by the LD

***lFDI attribute*** *(HexBinary160) [0..1]*

This element MUST be set to the LFDI of the device for which this file in targeted.

***mfHwVer attribute*** *(String32) [0..1]*

This element MUST be set to the hardware version for which this file is targeted.

***mfID attribute*** *(PENType)*

This element MUST be set to the manufacturer's Private Enterprise Number (assigned by IANA).

***mfModel attribute*** *(String32)*

This element MUST be set to the manufacturer model number for which this file is targeted. The syntax and semantics are left to the manufacturer.

***mfSerNum attribute*** *(String32) [0..1]*

This element MUST be set to the manufacturer serial number for which this file is targeted. The syntax and semantics are left to the manufacturer.

***mfVer attribute*** *(String16)*

This element MUST be set to the software version information for this file. The syntax and semantics are left to the manufacturer.

***size attribute*** *(UInt32)*

This element MUST be set to the total size (in bytes) of the file referenced by fileURI.

***type attribute*** *(HexBinary16)*

A value indicating the type of the file. MUST be one of the following values:

00 = Software Image

01 = Security Credential

02 = Configuration

03 = Log

04–7FFF = reserved

8000-FFFF = Manufacturer defined

**FileList Object** (List)

A List element to hold File objects.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

**FileStatus Object** (Resource)

This object provides status of device file load and activation operations.

***activateTime attribute*** *(TimeType) [0..1]*

Date/time at which this File, referred to by FileLink, will be activated. Omission of or presence and value of this element MUST exactly match omission or presence and value of the activateTime element from the File resource.

***loadPercent attribute*** *(UInt8)*

This element MUST be set to the percentage of the file, indicated by FileLink, that was loaded during the latest load attempt. This value MUST be reset to 0 each time a load attempt is started for the File indicated by FileLink. This value MUST be increased when an LD receives HTTP response containing file content. This value MUST be set to 100 when the full content of the file has been received by the LD

***nextRequestAttempt attribute*** *(TimeType)*

This element MUST be set to the time at which the LD will issue its next GET request for file content from the File indicated by FileLink

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

***request503Count attribute*** *(UInt16)*

This value MUST be reset to 0 when FileLink is first pointed at a new File. This value MUST be incremented each time an

LD receives a 503 error from the FS.

***requestFailCount attribute*** *(UInt16)*

This value MUST be reset to 0 when FileLink is first pointed at a new File. This value MUST be incremented each time a GET request for file content failed. 503 errors MUST be excluded from this counter.

***status attribute*** *(UInt8)*

Current loading status of the file indicated by FileLink. This element MUST be set to one of the following values:

0 - No load operation in progress

1 - File load in progress (first request for file content has been issued by LD)

2 - File load failed

3 - File loaded successfully (full content of file has been received by the LD), signature verification in progress

4 - File signature verification failed

5 - File signature verified, waiting to activate file.

6 - File activation failed

7 - File activation in progress

8 - File activated successfully (this state may not be reached/persisted through an image activation)

9-255 - Reserved for future use.

***statusTime attribute*** *(TimeType)*

This element MUST be set to the time at which file status transitioned to the value indicated in the status element.

### **DRLC Package**

Contains definitions for Demand Response Load Control functionality.



**Figure B.Diagram Sequence**—DRLC Event



**Figure B.Diagram Sequence**—Load Shed Availability

**LoadShedAvailabilityList Object** (List)

A List element to hold LoadShedAvailability objects.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

**ApplianceLoadReduction Object** ()

The ApplianceLoadReduction object is used by a Demand Response service provider to provide signals for ENERGY STAR compliant appliances. See the definition of ApplianceLoadReductionType for more information.

***type attribute*** *(ApplianceLoadReductionType)*

Indicates the type of appliance load reduction requested.

**DemandResponseProgram Object** (IdentifiedObject)

Demand response program.

***availabilityUpdatePercentChangeThreshold attribute*** *(PerCent) [0..1]*

This attribute allows program providers to specify the requested granularity of updates to LoadShedAvailability sheddablePercent. If not present, or set to 0, then updates to LoadShedAvailability SHALL NOT be provided. If present and greater than zero, then clients SHALL provide their LoadShedAvailability if it has not previously been provided, and thereafter if the difference between the previously provided value and the current value of LoadShedAvailability sheddablePercent is greater than availabilityUpdatePercentChangeThreshold.

***availabilityUpdatePowerChangeThreshold attribute*** *(ActivePower) [0..1]*

This attribute allows program providers to specify the requested granularity of updates to LoadShedAvailability sheddablePower. If not present, or set to 0, then updates to LoadShedAvailability SHALL NOT be provided. If present and greater than zero, then clients SHALL provide their LoadShedAvailability if it has not previously been provided, and thereafter if the difference between the previously provided value and the current value of LoadShedAvailability sheddablePower is greater than availabilityUpdatePowerChangeThreshold.

***primacy attribute*** *(PrimacyType)*

Indicates the relative primacy of the provider of this program.

**DemandResponseProgramList Object** (SubscribableList)

A List element to hold DemandResponseProgram objects.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

**DutyCycle Object** ()

Duty cycle control is a device specific issue and is managed by the device. The duty cycle of the device under control should span the shortest practical time period in accordance with the nature of the device under control and the intent of the request for demand reduction. The default factory setting SHOULD be three minutes for each 10% of duty cycle. This indicates that the default time period over which a duty cycle is applied is 30 minutes, meaning a 10% duty cycle would cause a device to be ON for 3 minutes. The “off state” SHALL precede the “on state”.

***normalValue attribute*** *(UInt8)*

Contains the maximum On state duty cycle applied by the end device, as a percentage of time. The field not present indicates that this field has not been used by the end device.

**EndDeviceControl Object** (RandomizableEvent)

Instructs an EndDevice to perform a specified action.

***deviceCategory attribute*** *(DeviceCategoryType)*

Specifies the bitmap indicating the categories of devices that SHOULD respond. Devices SHOULD ignore events that do not indicate their device category.

***drProgramMandatory attribute*** *(boolean)*

A flag to indicate if the EndDeviceControl is considered a mandatory event as defined by the service provider issuing the EndDeviceControl. The drProgramMandatory flag alerts the client/user that they will be subject to penalty or ineligibility based on the service provider’s program rules for that deviceCategory.

***loadShiftForward attribute*** *(boolean)*

Indicates that the event intends to increase consumption. A value of true indicates the intention to increase usage value, and a value of false indicates the intention to decrease usage.

***overrideDuration attribute*** *(UInt16) [0..1]*

The overrideDuration attribute provides a duration, in seconds, for which a client device is allowed to override this EndDeviceControl and still meet the contractual agreement with a service provider without opting out. If overrideDuration is not specified, then it SHALL default to 0.

**EndDeviceControlList Object** (SubscribableList)

A List element to hold EndDeviceControl objects.

**LoadShedAvailability Object** (Resource)

Indicates current consumption status and ability to shed load.

***availabilityDuration attribute*** *(UInt32) [0..1]*

Indicates for how many seconds the consuming device will be able to reduce consumption at the maximum response level.

***sheddablePercent attribute*** *(PerCent) [0..1]*

Maximum percent of current operating load that is estimated to be sheddable.

***sheddablePower attribute*** *(ActivePower) [0..1]*

Maximum amount of current operating load that is estimated to be sheddable, in Watts.

**Association Object** ()

**Offset Object** ()

If a temperature offset is sent that causes the heating or cooling temperature set point to exceed the limit boundaries that are programmed into the device, the device SHALL respond by setting the temperature at the limit.

If an EDC is being targeted at multiple devices or to a device that controls multiple devices (e.g., EMS), it can provide multiple Offset types within one EDC. For events with multiple Offset types, a client SHALL select the Offset that best fits their operating function.

Alternatively, an event with a single Offset type can be targeted at an EMS in order to request a percentage load reduction on the average energy usage of the entire premise. An EMS SHOULD use the Metering function set to determine the initial load in the premise, reduce energy consumption by controlling devices at its disposal, and at the conclusion of the event, once again use the Metering function set to determine if the desired load reduction was achieved.

***coolingOffset attribute*** *(UInt8) [0..1]*

The value change requested for the cooling offset, in degree C / 10. The value should be added to the normal set point for cooling, or if loadShiftForward is true, then the value should be subtracted from the normal set point.

***heatingOffset attribute*** *(UInt8) [0..1]*

The value change requested for the heating offset, in degree C / 10. The value should be subtracted for heating, or if loadShiftForward is true, then the value should be added to the normal set point.

***loadAdjustmentPercentageOffset attribute*** *(PerCent) [0..1]*

The value change requested for the load adjustment percentage. The value should be subtracted from the normal setting, or if loadShiftForward is true, then the value should be added to the normal setting.

**SetPoint Object** ()

The SetPoint object is used to apply specific temperature set points to a temperature control device. The values of the heatingSetpoint and coolingSetpoint attributes SHALL be calculated as follows:

Cooling/Heating Temperature Set Point / 100 = temperature in degrees Celsius where -273.15°C <= temperature <= 327.67°C, corresponding to a Cooling and/or Heating Temperature Set Point. The maximum resolution this format allows is 0.01°C.

The field not present in a Response indicates that this field has not been used by the end device.

If a temperature is sent that exceeds the temperature limit boundaries that are programmed into the device, the device SHALL respond by setting the temperature at the limit.

***coolingSetpoint attribute*** *(Int16) [0..1]*

This attribute represents the cooling temperature set point in degrees Celsius / 100. (Hundredths of a degree C)

***heatingSetpoint attribute*** *(Int16) [0..1]*

This attribute represents the heating temperature set point in degrees Celsius / 100. (Hundredths of a degree C)

**TargetReduction Object** ()

The TargetReduction object is used by a Demand Response service provider to provide a RECOMMENDED threshold that a device/premises should maintain its consumption below. For example, a service provider can provide a RECOMMENDED threshold of some kWh for a 3-hour event. This means that the device/premises would maintain its consumption below the specified limit for the specified period.

***type attribute*** *(UnitType)*

Indicates the type of reduction requested.

***value attribute*** *(UInt16)*

Indicates the requested amount of the relevant commodity to be reduced.

### **Metering Package**

Contains definitions related to measurements of energy at usage points.



**Figure B.Diagram Sequence**—Metering Data



**Figure B.Diagram Sequence**—Metering Data Types

**MeterReading Object** (MeterReadingBase)

Set of values obtained from the meter.

**MeterReadingList Object** (SubscribableList)

A List element to hold MeterReading objects.

**Reading Object** (ReadingBase)

Specific value measured by a meter or other asset.

***localID attribute*** *(HexBinary16) [0..1]*

The local identifier for this reading within the reading set. localIDs are assigned in order of creation time. For interval data, this value SHALL increase with each interval time, and for block/tier readings, localID SHALL not be specified.

***subscribable attribute*** *(SubscribableType) [0..1] «XSDattribute»*

Indicates whether or not subscriptions are supported for this resource, and whether or not conditional (thresholds) are supported. If not specified, is "not subscribable" (0).

**ReadingList Object** (SubscribableList)

A List element to hold Reading objects.

**ReadingSet Object** (ReadingSetBase)

A set of Readings of the ReadingType indicated by the parent MeterReading.

**ReadingSetList Object** (SubscribableList)

A List element to hold ReadingSet objects.

**ReadingType Object** (Resource)

Type of data conveyed by a specific Reading. See IEC 61968 Part 9 Annex C for full definitions of these values.

***accumulationBehaviour attribute*** *(AccumulationBehaviourType) [0..1]*

The “accumulation behaviour” indicates how the value is represented to accumulate over time.

***calorificValue attribute*** *(UnitValueType) [0..1]*

The amount of heat generated when a given mass of fuel is completely burned. The CalorificValue is used to convert the measured volume or mass of gas into kWh. The CalorificValue attribute represents the current active value.

***commodity attribute*** *(CommodityType) [0..1]*

Indicates the commodity applicable to this ReadingType.

***conversionFactor attribute*** *(UnitValueType) [0..1]*

Accounts for changes in the volume of gas based on temperature and pressure. The ConversionFactor attribute represents the current active value. The ConversionFactor is dimensionless. The default value for the ConversionFactor is 1, which means no conversion is applied. A price server can advertise a new/different value at any time.

***dataQualifier attribute*** *(DataQualifierType) [0..1]*

The data type can be used to describe a salient attribute of the data. Possible values are average, absolute, and etc.

***flowDirection attribute*** *(FlowDirectionType) [0..1]*

Anything involving current might have a flow direction. Possible values include forward and reverse.

***intervalLength attribute*** *(UInt32) [0..1]*

Default interval length specified in seconds.

***kind attribute*** *(KindType) [0..1]*

Compound class that contains kindCategory and kindIndex

***maxNumberOfIntervals attribute*** *(UInt8) [0..1]*

To be populated for mirrors of interval data to set the expected number of intervals per ReadingSet. Servers may discard intervals received that exceed this number.

***numberOfConsumptionBlocks attribute*** *(UInt8) [0..1]*

Number of consumption blocks. 0 means not applicable, and is the default if not specified. The value needs to be at least 1 if any actual prices are provided.

***numberOfTouTiers attribute*** *(UInt8) [0..1]*

The number of TOU tiers that can be used by any resource configured by this ReadingType. Servers SHALL populate this value with the largest touTier value that will *ever* be used while this ReadingType is in effect. Servers SHALL set numberOfTouTiers equal to the number of standard TOU tiers plus the number of CPP tiers that may be used while this ReadingType is in effect. Servers SHALL specify a value between 0 and 255 (inclusive) for numberOfTouTiers (servers providing flat rate pricing SHOULD set numberOfTouTiers to 0, as in practice there is no difference between having no tiers and having one tier).

***phase attribute*** *(PhaseCode) [0..1]*

Contains phase information associated with the type.

***powerOfTenMultiplier attribute*** *(PowerOfTenMultiplierType) [0..1]*

Indicates the power of ten multiplier applicable to the unit of measure of this ReadingType.

***subIntervalLength attribute*** *(UInt32) [0..1]*

Default sub-interval length specified in seconds for Readings of ReadingType. Some demand calculations are done over a number of smaller intervals. For example, in a rolling demand calculation, the demand value is defined as the rolling sum of smaller intervals over the intervalLength. The subintervalLength is the length of the smaller interval in this calculation. It SHALL be an integral division of the intervalLength. The number of sub-intervals can be calculated by dividing the intervalLength by the subintervalLength.

***supplyLimit attribute*** *(UInt48) [0..1]*

Reflects the supply limit set in the meter. This value can be compared to the Reading value to understand if limits are being approached or exceeded. Units follow the same definition as in this ReadingType.

***tieredConsumptionBlocks attribute*** *(boolean) [0..1]*

Specifies whether or not the consumption blocks are differentiated by TOUTier or not. Default is false, if not specified.

true = consumption accumulated over individual tiers

false = consumption accumulated over all tiers

***uom attribute*** *(UomType) [0..1]*

Indicates the measurement type for the units of measure for the readings of this type.

**UsagePoint Object** (UsagePointBase)

Logical point on a network at which consumption or production is either physically measured (e.g. metered) or estimated (e.g. unmetered street lights).

***deviceLFDI attribute*** *(HexBinary160) [0..1]*

The LFDI of the source device. This attribute SHALL be present when mirroring.

**UsagePointList Object** (SubscribableList)

A List element to hold UsagePoint objects.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

#### **Metering Mirror Package**



**Figure B.Diagram Sequence**—Metering Mirror



**Figure B.Diagram Sequence**—Metering Mirror Inheritance

**MirrorMeterReading Object** (MeterReadingBase)

Mimic of MeterReading used for managing mirrors.

***lastUpdateTime attribute*** *(TimeType) [0..1]*

The date and time of the last update.

***nextUpdateTime attribute*** *(TimeType) [0..1]*

The date and time of the next planned update.

**MirrorMeterReadingList Object** (List)

A List of MirrorMeterReading instances.

**MeterReadingBase Object** (IdentifiedObject)

A container for associating ReadingType, Readings and ReadingSets.

**MirrorReadingSet Object** (ReadingSetBase)

A set of Readings of the ReadingType indicated by the parent MeterReading.

**MirrorUsagePoint Object** (UsagePointBase)

A parallel to UsagePoint to support mirroring

***deviceLFDI attribute*** *(HexBinary160)*

The LFDI of the device being mirrored.

***postRate attribute*** *(UInt32) [0..1]*

POST rate, or how often mirrored data should be POSTed, in seconds. A client MAY indicate a preferred postRate when POSTing MirrorUsagePoint. A server MAY add or modify postRate to indicate its preferred posting rate.

**MirrorUsagePointList Object** (List)

A List of MirrorUsagePoint instances.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

**ReadingBase Object** (Resource)

Specific value measured by a meter or other asset. ReadingBase is abstract, used to define the elements common to Reading and IntervalReading.

***consumptionBlock attribute*** *(ConsumptionBlockType) [0..1]*

Indicates the consumption block related to the reading. REQUIRED if ReadingType numberOfConsumptionBlocks is non-zero. If not specified, is assumed to be "0 - N/A".

***qualityFlags attribute*** *(HexBinary16) [0..1]*

List of codes indicating the quality of the reading, using specification:

Bit 0 - valid: data that has gone through all required validation checks and either passed them all or has been verified

Bit 1 - manually edited: Replaced or approved by a human

Bit 2 - estimated using reference day: data value was replaced by a machine computed value based on analysis of historical data using the same type of measurement.

Bit 3 - estimated using linear interpolation: data value was computed using linear interpolation based on the readings before and after it

Bit 4 - questionable: data that has failed one or more checks

Bit 5 - derived: data that has been calculated (using logic or mathematical operations), not necessarily measured directly

Bit 6 - projected (forecast): data that has been calculated as a projection or forecast of future readings

***timePeriod attribute*** *(DateTimeInterval) [0..1]*

The time interval associated with the reading. If not specified, then defaults to the intervalLength specified in the associated ReadingType.

***touTier attribute*** *(TOUType) [0..1]*

Indicates the time of use tier related to the reading. REQUIRED if ReadingType numberOfTouTiers is non-zero. If not specified, is assumed to be "0 - N/A".

***value attribute*** *(Int48) [0..1]*

Value in units specified by ReadingType

**ReadingSetBase Object** (IdentifiedObject)

A set of Readings of the ReadingType indicated by the parent MeterReading. ReadingBase is abstract, used to define the elements common to ReadingSet and IntervalBlock.

***timePeriod attribute*** *(DateTimeInterval)*

Specifies the time range during which the contained readings were taken.

**UsagePointBase Object** (IdentifiedObject)

Logical point on a network at which consumption or production is either physically measured (e.g. metered) or estimated (e.g. unmetered street lights). A container for associating ReadingType, Readings and ReadingSets.

***roleFlags attribute*** *(RoleFlagsType)*

Specifies the roles that apply to the usage point.

***serviceCategoryKind attribute*** *(ServiceKind)*

The kind of service provided by this usage point.

***status attribute*** *(UInt8)*

Specifies the current status of the service at this usage point.

0 = off

1 = on

### **Pricing Package**

Contains definitions of information related to price.



**Figure B.Diagram Sequence**—Pricing

**ConsumptionTariffInterval Object** (Resource)

One of a sequence of thresholds defined in terms of consumption quantity of a service such as electricity, water, gas, etc. It defines the steps or blocks in a step tariff structure, where startValue simultaneously defines the entry value of this step and the closing value of the previous step. Where consumption is greater than startValue, it falls within this block and where consumption is less than or equal to startValue, it falls within one of the previous blocks.

***consumptionBlock attribute*** *(ConsumptionBlockType)*

Indicates the consumption block related to the reading. If not specified, is assumed to be "0 - N/A".

***price attribute*** *(Int32) [0..1]*

The charge for this rate component, per unit of measure defined by the associated ReadingType, in currency specified in TariffProfile.

The Pricing service provider determines the appropriate price attribute value based on its applicable regulatory rules. For example, price could be net or inclusive of applicable taxes, fees, or levies.

The Billing function set provides the ability to represent billing information in a more detailed manner.

***startValue attribute*** *(UInt48)*

The lowest level of consumption that defines the starting point of this consumption step or block. Thresholds start at zero for each billing period.

If specified, the first ConsumptionTariffInterval.startValue for a TimeTariffInteral instance SHALL begin at "0." Subsequent ConsumptionTariffInterval.startValue elements SHALL be greater than the previous one.

**ConsumptionTariffIntervalList Object** (List)

A List element to hold ConsumptionTariffInterval objects.

**CostKindType Object** (UInt8)

0 - Carbon Dioxide emissions, in grams per unit

1 - Sulfur Dioxide emissions, in grams per unit

2 - Nitrogen Oxides emissions, in grams per unit

3 - Renewable generation, as a percentage of overall generation

All other values reserved.

**EnvironmentalCost Object** ()

Provides alternative or secondary price information for the relevant RateComponent. Supports jurisdictions that seek to convey the environmental price per unit of the specified commodity not expressed in currency.

Implementers and consumers can use this attribute to prioritize operations of their HAN devices (e.g., PEV charging during times of high availability of renewable electricity resources).

***amount attribute*** *(UInt32)*

The estimated or actual environmental or other cost, per commodity unit defined by the ReadingType, for this RateComponent (e.g., grams of carbon dioxide emissions each per kWh).

***costKind attribute*** *(CostKindType)*

The kind of cost referred to in the amount.

***costLevel attribute*** *(UInt8)*

The relative level of the amount attribute. In conjunction with numCostLevels, this attribute informs a device of the relative scarcity of the amount attribute (e.g., a high or low availability of renewable generation).

numCostLevels and costLevel values SHALL ascend in order of scarcity, where "0" signals the lowest relative cost and higher values signal increasing cost. For example, if numCostLevels is equal to “3,” then if the lowest relative costLevel were equal to “0,” devices would assume this is the lowest relative period to operate. Likewise, if the costLevel in the next TimeTariffInterval instance is equal to “1,” then the device would assume it is relatively more expensive, in environmental terms, to operate during this TimeTariffInterval instance than the previous one.

There is no limit to the number of relative price levels other than that indicated in the attribute type, but for practicality, service providers should strive for simplicity and recognize the diminishing returns derived from increasing the numCostLevel value greater than four.

***numCostLevels attribute*** *(UInt8)*

The number of all relative cost levels.

In conjunction with costLevel, numCostLevels signals the relative scarcity of the commodity for the duration of the TimeTariffInterval instance (e.g., a relative indication of cost). This is useful in providing context for nominal cost signals to consumers or devices that might see a range of amount values from different service providres or from the same service provider.

**RateComponent Object** (IdentifiedObject)

Specifies the applicable charges for a single component of the rate, which could be generation price or consumption price, for example.

***flowRateEndLimit attribute*** *(UnitValueType) [0..1]*

Specifies the maximum flow rate (e.g. kW for electricity) for which this RateComponent applies, for the usage point and given rate / tariff.

In combination with flowRateStartLimit, allows a service provider to define the demand or output characteristics for the particular tariff design. If a server includes the flowRateEndLimit attribute, then it SHALL also include flowRateStartLimit attribute.

For example, a service provider’s tariff limits customers to 20 kWs of demand for the given rate structure. Above this threshold (from 20-50 kWs), there are different demand charges per unit of consumption. The service provider can use flowRateStartLimit and flowRateEndLimit to describe the demand characteristics of the different rates. Similarly, these attributes can be used to describe limits on premises DERs that might be producing a commodity and sending it back into the distribution network.

Note: At the time of writing, service provider tariffs with demand-based components were not originally identified as being in scope, and service provider tariffs vary widely in their use of demand components and the method for computing charges. It is expected that industry groups (e.g., OpenSG) will document requirements in the future that the IEEE 2030.5 community can then use as source material for the next version of IEEE 2030.5.

***flowRateStartLimit attribute*** *(UnitValueType) [0..1]*

Specifies the minimum flow rate (e.g., kW for electricity) for which this RateComponent applies, for the usage point and given rate / tariff.

In combination with flowRateEndLimit, allows a service provider to define the demand or output characteristics for the particular tariff design. If a server includes the flowRateStartLimit attribute, then it SHALL also include flowRateEndLimit attribute.

***roleFlags attribute*** *(RoleFlagsType)*

Specifies the roles that this usage point has been assigned.

**RateComponentList Object** (List)

A List element to hold RateComponent objects.

**TariffProfile Object** (IdentifiedObject)

A schedule of charges; structure that allows the definition of tariff structures such as step (block) and time of use (tier) when used in conjunction with TimeTariffInterval and ConsumptionTariffInterval.

***currency attribute*** *(CurrencyCode) [0..1]*

The currency code indicating the currency for this TariffProfile.

***pricePowerOfTenMultiplier attribute*** *(PowerOfTenMultiplierType) [0..1]*

Indicates the power of ten multiplier for the price attribute.

***primacy attribute*** *(PrimacyType)*

Indicates the relative primacy of the provider of this program.

***rateCode attribute*** *(String20) [0..1]*

The rate code for this tariff profile. Provided by the Pricing service provider per its internal business needs and practices and provides a method to identify the specific rate code for the TariffProfile instance. This would typically not be communicated to the user except to facilitate troubleshooting due to its service provider-specific technical nature.

***serviceCategoryKind attribute*** *(ServiceKind)*

The kind of service provided by this usage point.

**TariffProfileList Object** (SubscribableList)

A List element to hold TariffProfile objects.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

**TimeTariffInterval Object** (RandomizableEvent)

Describes the time-differentiated portion of the RateComponent, if applicable, and provides the ability to specify multiple time intervals, each with its own consumption-based components and other attributes.

***touTier attribute*** *(TOUType)*

Indicates the time of use tier related to the reading. If not specified, is assumed to be "0 - N/A".

**TimeTariffIntervalList Object** (SubscribableList)

A List element to hold TimeTariffInterval objects.

### **Messaging Package**

Contains text message definitions.



**Figure B.Diagram Sequence**—TextMessage

**MessagingProgram Object** (SubscribableIdentifiedObject)

Provides a container for collections of text messages.

***locale attribute*** *(LocaleType)*

Indicates the language and region of the messages in this collection.

***primacy attribute*** *(PrimacyType)*

Indicates the relative primacy of the provider of this program.

**MessagingProgramList Object** (SubscribableList)

A List element to hold MessagingProgram objects.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

**PriorityType Object** (UInt8)

Indicates the priority of a message:

0 - Low

1 - Normal

2 - High

3 - Critical

All other values reserved.

**TextMessage Object** (Event)

Text message such as a notification.

***originator attribute*** *(String20) [0..1]*

Indicates the human-readable name of the publisher of the message

***priority attribute*** *(PriorityType)*

The priority is used to inform the client of the priority of the particular message. Devices with constrained or limited resources for displaying Messages should use this attribute to determine how to handle displaying currently active Messages (e.g. if a device uses a scrolling method with a single Message viewable at a time it MAY want to push a low priority Message to the background and bring a newly received higher priority Message to the foreground).

***textMessage attribute*** *(string)*

The textMessage attribute contains the actual UTF-8 encoded text to be displayed in conjunction with the messageLength attribute which contains the overall length of the textMessage attribute. Clients and servers SHALL support a reception of a Message of 100 bytes in length. Messages that exceed the clients display size will be left to the client to choose what method to handle the message (truncation, scrolling, etc.).

**TextMessageList Object** (SubscribableList)

A List element to hold TextMessage objects.

### **Billing Package**

Contains representations of charges and other billing related information.



**Figure B.Diagram Sequence**—Billing



**Figure B.Diagram Sequence**—Billing Readings

**BillingPeriod Object** (Resource)

A Billing Period relates to the period of time on which a customer is billed. As an example the billing period interval for a particular customer might be 31 days starting on July 1, 2011. The start date and interval can change on each billing period. There may also be multiple billing periods related to a customer agreement to support different tariff structures.

***billLastPeriod attribute*** *(Int48) [0..1]*

The amount of the bill for the previous billing period.

***billToDate attribute*** *(Int48) [0..1]*

The bill amount related to the billing period as of the statusTimeStamp.

***interval attribute*** *(DateTimeInterval)*

The time interval for this billing period.

***statusTimeStamp attribute*** *(TimeType) [0..1]*

The date / time of the last update of this resource.

**BillingPeriodList Object** (SubscribableList)

A List element to hold BillingPeriod objects.

**BillingMeterReadingBase Object** (MeterReadingBase)

Contains historical, target, and projection readings of various types, possibly associated with charges.

**BillingReading Object** (ReadingBase)

Data captured at regular intervals of time. Interval data could be captured as incremental data, absolute data, or relative data. The source for the data is usually a tariff quantity or an engineering quantity. Data is typically captured in time-tagged, uniform, fixed-length intervals of 5 min, 10 min, 15 min, 30 min, or 60 min. However, consumption aggregations can also be represented with this class.

**BillingReadingList Object** (List)

A List element to hold BillingReading objects.

**BillingReadingSet Object** (ReadingSetBase)

Time sequence of readings of the same reading type.

**BillingReadingSetList Object** (SubscribableList)

A List element to hold BillingReadingSet objects.

**Charge Object** ()

Charges contain charges on a customer bill. These could be items like taxes, levies, surcharges, rebates, or others. This is meant to allow the HAN device to retrieve enough information to be able to reconstruct an estimate of what the total bill would look like.

Providers can provide line item billing, including multiple charge kinds (e.g. taxes, surcharges) at whatever granularity desired, using as many Charges as desired during a billing period. There can also be any number of Charges associated with different ReadingTypes to distinguish between TOU tiers, consumption blocks, or demand charges.

***description attribute*** *(String20) [0..1]*

A description of the charge.

***kind attribute*** *(ChargeKind) [0..1]*

The type (kind) of charge.

***value attribute*** *(Int32)*

A monetary charge.

**ChargeKind Object** (UInt8)

Kind of charge.

0 - Consumption Charge

1 - Rebate

2 - Auxiliary Charge

3 - Demand Charge

4 - Tax Charge

**CustomerAccount Object** (IdentifiedObject)

Assignment of a group of products and services purchased by the Customer through a CustomerAgreement, used as a mechanism for customer billing and payment. It contains common information from the various types of CustomerAgreements to create billings (invoices) for a Customer and receive payment.

***currency attribute*** *(UInt16)*

The ISO 4217 code indicating the currency applicable to the bill amounts in the summary. See list at http://www.unece.org/cefact/recommendations/rec09/rec09\_ecetrd203.pdf

***customerAccount attribute*** *(String42) [0..1]*

The account number for the customer (if applicable).

***customerName attribute*** *(String42) [0..1]*

The name of the customer.

***pricePowerOfTenMultiplier attribute*** *(PowerOfTenMultiplierType)*

Indicates the power of ten multiplier for the prices in this function set.

**CustomerAccountList Object** (SubscribableList)

A List element to hold CustomerAccount objects.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

**CustomerAgreement Object** (IdentifiedObject)

Agreement between the customer and the service supplier to pay for service at a specific service location. It records certain billing information about the type of service provided at the service location and is used during charge creation to determine the type of service.

***serviceAccount attribute*** *(String42) [0..1]*

The account number of the service account (if applicable).

***serviceLocation attribute*** *(String42) [0..1]*

The address or textual description of the service location.

**CustomerAgreementList Object** (SubscribableList)

A List element to hold CustomerAgreement objects.

**HistoricalReading Object** (BillingMeterReadingBase)

To be used to present readings that have been processed and possibly corrected (as allowed, due to missing or incorrect data) by backend systems. This includes quality codes valid, verified, estimated, and derived / corrected.

**HistoricalReadingList Object** (List)

A List element to hold HistoricalReading objects.

**ProjectionReading Object** (BillingMeterReadingBase)

Contains values that forecast a future reading for the time or interval specified.

**ProjectionReadingList Object** (List)

A List element to hold ProjectionReading objects.

**TargetReading Object** (BillingMeterReadingBase)

Contains readings that specify a target or goal, such as a consumption target, to which billing incentives or other contractual ramifications may be associated.

**TargetReadingList Object** (List)

A List element to hold TargetReading objects.

**ServiceSupplier Object** (IdentifiedObject)

Organisation that provides services to Customers.

***email attribute*** *(String32) [0..1]*

E-mail address for this service supplier.

***phone attribute*** *(String20) [0..1]*

Human-readable phone number for this service supplier.

***providerID attribute*** *(UInt32) [0..1]*

Contains the IANA PEN for the commodity provider.

***web attribute*** *(String42) [0..1]*

Website URI address for this service supplier.

**ServiceSupplierList Object** (List)

A List element to hold ServiceSupplier objects.

### **Prepayment Package**

Contains definitions related to storing and using payments.



**Figure B.Diagram Sequence**—Prepayment

**AccountBalance Object** (Resource)

AccountBalance contains the regular credit and emergency credit balance for this given service or commodity prepay instance. It may also contain status information concerning the balance data.

***availableCredit attribute*** *(AccountingUnit)*

AvailableCredit shows the balance of the sum of credits minus the sum of charges. In a Central Wallet mode this value may be passed down to the Prepayment server via an out-of-band mechanism. In Local or ESI modes, this value may be calculated based upon summation of CreditRegister transactions minus consumption charges calculated using Metering (and possibly Pricing) function set data. This value may be negative; for instance, if disconnection is prevented due to a Supply Interruption Override.

***creditStatus attribute*** *(CreditStatusType) [0..1]*

CreditStatus identifies whether the present value of availableCredit is considered OK, low, exhausted, or negative.

***emergencyCredit attribute*** *(AccountingUnit) [0..1]*

EmergencyCredit is the amount of credit still available for the given service or commodity prepayment instance. If both availableCredit and emergyCredit are exhausted, then service will typically be disconnected.

***emergencyCreditStatus attribute*** *(CreditStatusType) [0..1]*

EmergencyCreditStatus identifies whether the present value of emergencyCredit is considered OK, low, exhausted, or negative.

**AccountingUnit Object** ()

Unit for accounting; use either 'energyUnit' or 'currencyUnit' to specify the unit for 'value'.

***energyUnit attribute*** *(RealEnergy) [0..1]*

Unit of service.

***monetaryUnit attribute*** *(CurrencyCode)*

Unit of currency.

***multiplier attribute*** *(PowerOfTenMultiplierType)*

Multiplier for the 'energyUnit' or 'monetaryUnit'.

***value attribute*** *(Int32)*

Value of the monetary aspect

**CreditRegister Object** (IdentifiedObject)

CreditRegister instances define a credit-modifying transaction. Typically this would be a credit-adding transaction, but may be a subtracting transaction (perhaps in response to an out-of-band debt signal).

***creditAmount attribute*** *(AccountingUnit)*

CreditAmount is the amount of credit being added by a particular CreditRegister transaction. Negative values indicate that credit is being subtracted.

***creditType attribute*** *(CreditTypeType) [0..1]*

CreditType indicates whether the credit transaction applies to regular or emergency credit.

***effectiveTime attribute*** *(TimeType)*

EffectiveTime identifies the time at which the credit transaction goes into effect. For credit addition transactions, this is typically the moment at which the transaction takes place. For credit subtraction transactions, (e.g., non-fuel debt recovery transactions initiated from a back-haul or ESI) this may be a future time at which credit is deducted.

***token attribute*** *(String32)*

Token is security data that authenticates the legitimacy of the transaction. The details of this token are not defined by IEEE 2030.5. How a Prepayment server handles this field is left as vendor specific implementation or will be defined by one or more other standards.

**CreditRegisterList Object** (List)

A List element to hold CreditRegister objects.

**Prepayment Object** (IdentifiedObject)

Prepayment (inherited from CIM SDPAccountingFunction)

***creditExpiryLevel attribute*** *(AccountingUnit) [0..1]*

CreditExpiryLevel is the set point for availableCredit at which the service level may be changed. The typical value for this attribute is 0, regardless of whether the account balance is measured in a monetary or commodity basis. The units for this attribute SHALL match the units used for availableCredit.

***lowCreditWarningLevel attribute*** *(AccountingUnit) [0..1]*

LowCreditWarningLevel is the set point for availableCredit at which the creditStatus attribute in the AccountBalance resource SHALL indicate that available credit is low. The units for this attribute SHALL match the units used for availableCredit. Typically, this value is set by the service provider.

***lowEmergencyCreditWarningLevel attribute*** *(AccountingUnit) [0..1]*

LowEmergencyCreditWarningLevel is the set point for emergencyCredit at which the creditStatus attribute in the AccountBalance resource SHALL indicate that emergencycredit is low. The units for this attribute SHALL match the units used for availableCredit. Typically, this value is set by the service provider.

***prepayMode attribute*** *(PrepayModeType)*

PrepayMode specifies whether the given Prepayment instance is operating in Credit, Central Wallet, ESI, or Local prepayment mode. The Credit mode indicates that prepayment is not presently in effect. The other modes are described in the Overview Section above.

**PrepaymentList Object** (SubscribableList)

A List element to hold Prepayment objects.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

**PrepayModeType Object** (UInt8)

0 - Central Wallet

1 - ESI

2 - Local

3 - Credit

All other values reserved.

**PrepayOperationStatus Object** (Resource)

PrepayOperationStatus describes the status of the service or commodity being conditionally controlled by the Prepayment function set.

***creditTypeChange attribute*** *(CreditTypeChange) [0..1]*

CreditTypeChange is used to define a pending change of creditTypeInUse, which will activate at a specified time.

***creditTypeInUse attribute*** *(CreditTypeType) [0..1]*

CreditTypeInUse identifies whether the present mode of operation is consuming regular credit or emergency credit.

***serviceChange attribute*** *(ServiceChange) [0..1]*

ServiceChange is used to define a pending change of serviceStatus, which will activate at a specified time.

***serviceStatus attribute*** *(ServiceStatusType)*

ServiceStatus identifies whether the service is connected or disconnected, or armed for connection or disconnection.

**ServiceChange Object** ()

Specifies a change to the service status.

***newStatus attribute*** *(ServiceStatusType)*

The new service status, to take effect at the time specified by startTime

***startTime attribute*** *(TimeType)*

The date/time when the change is to take effect.

**SupplyInterruptionOverride Object** (Resource)

SupplyInterruptionOverride: There may be periods of time when social, regulatory or other concerns mean that service should not be interrupted, even when available credit has been exhausted. Each Prepayment instance links to a List of SupplyInterruptionOverride instances. Each SupplyInterruptionOverride defines a contiguous period of time during which supply SHALL NOT be interrupted.

***description attribute*** *(String32) [0..1]*

The description is a human readable text describing or naming the object.

***interval attribute*** *(DateTimeInterval)*

Interval defines the period of time during which supply should not be interrupted.

**SupplyInterruptionOverrideList Object** (List)

A List element to hold SupplyInterruptionOverride objects.

**CreditStatusType Object** (UInt8)

0 - Credit Ok

1 - Credit Low

2 - Credit Exhausted

3 - Credit Negative

All other values reserved.

**CreditTypeType Object** (UInt8)

0 - Regular

1 - Emergency

2 - Regular, then Emergency

3 - Emergency, then Regular

All other values reserved.

**CreditTypeChange Object** ()

Specifies a change to the credit type.

***newType attribute*** *(CreditTypeType)*

The new credit type, to take effect at the time specified by startTime

***startTime attribute*** *(TimeType)*

The date/time when the change is to take effect.

**ServiceStatusType Object** (UInt8)

0 - Connected

1 - Disconnected

2 - Armed for Connect

3 - Armed for Disconnect

4 - No Contactor

5 - Load Limited

All other values reserved.

### **FlowReservation Package**

Contains flow (charge) reservation model to allow fine-grained control of high-demand loads such as fast-charging batteries.



**Figure B.Diagram Sequence**—FlowReservation

**RequestStatus Object** ()

The RequestStatus object is used to indicate the current status of a Flow Reservation Request.

***dateTime attribute*** *(TimeType)*

The dateTime attribute will provide a timestamp of when the request status was set. dateTime MUST be set to the time at which the status change occurred, not a time in the future or past.

***requestStatus attribute*** *(UInt8)*

Field representing the request status type.

0 = Requested

1 = Cancelled

All other values reserved.

**FlowReservationRequest Object** (IdentifiedObject)

Used to request flow transactions. Client EndDevices submit a request for charging or discharging from the server. The server creates an associated FlowReservationResponse containing the charging parameters and interval to provide a lower aggregated demand at the premises, or within a larger part of the distribution system.

***creationTime attribute*** *(TimeType)*

The time at which the request was created.

***durationRequested attribute*** *(UInt16) [0..1]*

A value that is calculated by the storage device that defines the minimum duration, in seconds, that it will take to complete the actual flow transaction, including any ramp times and conditioning times, if applicable.

***energyRequested attribute*** *(SignedRealEnergy)*

Indicates the total amount of energy, in Watt-Hours, requested to be transferred between the storage device and the electric power system. Positive values indicate charging and negative values indicate discharging. This sign convention is different than for the DER function where discharging is positive. Note that the energyRequestNow attribute in the PowerStatus Object must always represent a charging solution and it is not allowed to have a negative value.

***intervalRequested attribute*** *(DateTimeInterval)*

The time window during which the flow reservation is needed. For example, if an electric vehicle is set with a 7:00 AM time charge is needed, and price drops to the lowest tier at 11:00 PM, then this window would likely be from 11:00 PM until 7:00 AM.

***powerRequested attribute*** *(ActivePower)*

Indicates the sustained level of power, in Watts, that is requested. For charging this is calculated by the storage device and it represents the charging system capability (which for an electric vehicle must also account for any power limitations due to the EVSE control pilot). For discharging, a lower value than the inverter capability can be used as a target.

**FlowReservationRequestList Object** (List)

A List element to hold FlowReservationRequest objects.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

**FlowReservationResponse Object** (Event)

The server may modify the charging or discharging parameters and interval to provide a lower aggregated demand at the premises, or within a larger part of the distribution system.

***energyAvailable attribute*** *(SignedRealEnergy)*

Indicates the amount of energy available.

***powerAvailable attribute*** *(ActivePower)*

Indicates the amount of power available.

***subject attribute*** *(mRIDType)*

The subject field provides a method to match the response with the originating event. It is populated with the mRID of the corresponding FlowReservationRequest object.

**FlowReservationResponseList Object** (SubscribableList)

A List element to hold FlowReservationResponse objects.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

### **DER Package**

Contains definitions related to allowing distributed energy resources to provide energy back to the grid.



**Figure B.Diagram Sequence**—DER Info



**Figure B.Diagram Sequence**—DER Info Types



**Figure B.Diagram Sequence**—DER Program



**Figure B.Diagram Sequence**—DER Curves



**Figure B.Diagram Sequence**—DER Control



**Figure B.38**—DER Control Types

**DefaultDERControl Object** (SubscribableIdentifiedObject)

Contains control mode information to be used if no active DERControl is found.

***setESDelay attribute*** *(UInt32) [0..1]*

Enter service delay, in hundredths of a second. When present, this value SHALL update the value of the corresponding setting (DERSettings::setESDelay).

***setESHighFreq attribute*** *(UInt16) [0..1]*

Enter service frequency high. Specified in hundredths of Hz. When present, this value SHALL update the value of the corresponding setting (DERSettings::setESHighFreq).

***setESHighVolt attribute*** *(Int16) [0..1]*

Enter service voltage high. Specified as an effective percent voltage, defined as (100% \* (locally measured voltage - setVRefOfs) / setVRef), in hundredths of a percent. When present, this value SHALL update the value of the corresponding setting (DERSettings::setESHighVolt).

***setESLowFreq attribute*** *(UInt16) [0..1]*

Enter service frequency low. Specified in hundredths of Hz. When present, this value SHALL update the value of the corresponding setting (DERSettings::setESLowFreq).

***setESLowVolt attribute*** *(Int16) [0..1]*

Enter service voltage low. Specified as an effective percent voltage, defined as (100% \* (locally measured voltage - setVRefOfs) / setVRef), in hundredths of a percent. When present, this value SHALL update the value of the corresponding setting (DERSettings::setESLowVolt).

***setESRampTms attribute*** *(UInt32) [0..1]*

Enter service ramp time, in hundredths of a second. When present, this value SHALL update the value of the corresponding setting (DERSettings::setESRampTms).

***setESRandomDelay attribute*** *(UInt32) [0..1]*

Enter service randomized delay, in hundredths of a second. When present, this value SHALL update the value of the corresponding setting (DERSettings::setESRandomDelay).

***setGradW attribute*** *(UInt16) [0..1]*

Set default rate of change (ramp rate) of active power output due to command or internal action, defined in %setWMax / second. Resolution is in hundredths of a percent/second. A value of 0 means there is no limit. Interpreted as a percentage change in output capability limit per second when used as a default ramp rate. When present, this value SHALL update the value of the corresponding setting (DERSettings::setGradW).

***setSoftGradW attribute*** *(UInt16) [0..1]*

Set soft-start rate of change (soft-start ramp rate) of active power output due to command or internal action, defined in %setWMax / second. Resolution is in hundredths of a percent/second. A value of 0 means there is no limit. Interpreted as a percentage change in output capability limit per second when used as a ramp rate. When present, this value SHALL update the value of the corresponding setting (DERSettings::setSoftGradW).

**FreqDroopType Object** ()

Type for Frequency-Droop (Frequency-Watt) operation.

***dBOF attribute*** *(UInt32)*

Frequency droop dead band for over-frequency conditions. In thousandths of Hz.

***dBUF attribute*** *(UInt32)*

Frequency droop dead band for under-frequency conditions. In thousandths of Hz.

***kOF attribute*** *(UInt16)*

Frequency droop per-unit frequency change for over-frequency conditions corresponding to 1 per-unit power output change. In thousandths, unitless.

***kUF attribute*** *(UInt16)*

Frequency droop per-unit frequency change for under-frequency conditions corresponding to 1 per-unit power output change. In thousandths, unitless.

***openLoopTms attribute*** *(UInt16)*

Open loop response time, the duration from a step change in control signal input until the output changes by 90% of its final change before any overshoot, in hundredths of a second. Resolution is 1/100 sec. A value of 0 is used to mean no limit.

**DER Object** (SubscribableResource)

Contains links to DER resources.

**DERList Object** (List)

A List element to hold DER objects.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

**DERSettings Object** (SubscribableResource)

Distributed energy resource settings

***modesEnabled attribute*** *(DERControlType) [0..1]*

Bitmap indicating the DER Controls enabled on the device. See DERControlType for values. If a control is supported (see DERCapability::modesSupported), but not enabled, the control will not be executed if encountered.

***setESDelay attribute*** *(UInt32) [0..1]*

Enter service delay, in hundredths of a second.

***setESHighFreq attribute*** *(UInt16) [0..1]*

Enter service frequency high. Specified in hundredths of Hz.

***setESHighVolt attribute*** *(Int16) [0..1]*

Enter service voltage high. Specified as an effective percent voltage, defined as (100% \* (locally measured voltage - setVRefOfs) / setVRef), in hundredths of a percent.

***setESLowFreq attribute*** *(UInt16) [0..1]*

Enter service frequency low. Specified in hundredths of Hz.

***setESLowVolt attribute*** *(Int16) [0..1]*

Enter service voltage low. Specified as an effective percent voltage, defined as (100% \* (locally measured voltage - setVRefOfs) / setVRef), in hundredths of a percent.

***setESRampTms attribute*** *(UInt32) [0..1]*

Enter service ramp time, in hundredths of a second.

***setESRandomDelay attribute*** *(UInt32) [0..1]*

Enter service randomized delay, in hundredths of a second.

***setGradW attribute*** *(UInt16)*

Set default rate of change (ramp rate) of active power output due to command or internal action, defined in %setWMax / second. Resolution is in hundredths of a percent/second. A value of 0 means there is no limit. Interpreted as a percentage change in output capability limit per second when used as a default ramp rate.

***setMaxA attribute*** *(CurrentRMS) [0..1]*

AC current maximum. Maximum AC current in RMS Amperes.

***setMaxAh attribute*** *(AmpereHour) [0..1]*

Maximum usable energy storage capacity of the DER, in AmpHours. Note: this may be different from physical capability.

***setMaxChargeRateVA attribute*** *(ApparentPower) [0..1]*

Apparent power charge maximum. Maximum apparent power the DER can absorb from the grid in Volt-Amperes. May differ from the apparent power maximum (setMaxVA).

***setMaxChargeRateW attribute*** *(ActivePower) [0..1]*

Maximum rate of energy transfer received by the storage device, in Watts. Defaults to rtgMaxChargeRateW.

***setMaxDischargeRateVA attribute*** *(ApparentPower) [0..1]*

Apparent power discharge maximum. Maximum apparent power the DER can deliver to the grid in Volt-Amperes. May differ from the apparent power maximum (setMaxVA).

***setMaxDischargeRateW attribute*** *(ActivePower) [0..1]*

Maximum rate of energy transfer delivered by the storage device, in Watts. Defaults to rtgMaxDischargeRateW.

***setMaxV attribute*** *(VoltageRMS) [0..1]*

AC voltage maximum setting.

***setMaxVA attribute*** *(ApparentPower) [0..1]*

Set limit for maximum apparent power capability of the DER (in VA). Defaults to rtgMaxVA.

***setMaxVar attribute*** *(ReactivePower) [0..1]*

Set limit for maximum reactive power delivered by the DER (in var). SHALL be a positive value <= rtgMaxVar (default).

***setMaxVarNeg attribute*** *(ReactivePower) [0..1]*

Set limit for maximum reactive power received by the DER (in var). If present, SHALL be a negative value >= rtgMaxVarNeg (default). If absent, defaults to negative setMaxVar.

***setMaxW attribute*** *(ActivePower)*

Set limit for maximum active power capability of the DER (in W). Defaults to rtgMaxW.

***setMaxWh attribute*** *(WattHour) [0..1]*

Maximum energy storage capacity of the DER, in WattHours. Note: this may be different from physical capability.

***setMinPFOverExcited attribute*** *(PowerFactor) [0..1]*

Set minimum Power Factor displacement limit of the DER when injecting reactive power (over-excited); SHALL be a positive value between 0.0 (typically > 0.7) and 1.0. SHALL be >= rtgMinPFOverExcited (default).

***setMinPFUnderExcited attribute*** *(PowerFactor) [0..1]*

Set minimum Power Factor displacement limit of the DER when absorbing reactive power (under-excited); SHALL be a positive value between 0.0 (typically > 0.7) and 0.9999. If present, SHALL be >= rtgMinPFUnderExcited (default). If absent, defaults to setMinPFOverExcited.

***setMinV attribute*** *(VoltageRMS) [0..1]*

AC voltage minimum setting.

***setSoftGradW attribute*** *(UInt16) [0..1]*

Set soft-start rate of change (soft-start ramp rate) of active power output due to command or internal action, defined in %setWMax / second. Resolution is in hundredths of a percent/second. A value of 0 means there is no limit. Interpreted as a percentage change in output capability limit per second when used as a ramp rate.

***setVNom attribute*** *(VoltageRMS) [0..1]*

AC voltage nominal setting.

***setVRef attribute*** *(VoltageRMS) [0..1]*

The nominal AC voltage (RMS) at the utility's point of common coupling.

***setVRefOfs attribute*** *(VoltageRMS) [0..1]*

The nominal AC voltage (RMS) offset between the DER's electrical connection point and the utility's point of common coupling.

***updatedTime attribute*** *(TimeType)*

Specifies the time at which the DER information was last updated.

**DERType Object** (UInt8)

0 - Not applicable / Unknown

1 - Virtual or mixed DER

2 - Reciprocating engine

3 - Fuel cell

4 - Photovoltaic system

5 - Combined heat and power

6 - Other generation system

80 - Other storage system

81 - Electric vehicle

82 - EVSE

83 - Combined PV and storage

All other values reserved.

**DERAvailability Object** (SubscribableResource)

Indicates current reserve generation status

***availabilityDuration attribute*** *(UInt32) [0..1]*

Indicates number of seconds the DER will be able to deliver active power at the reservePercent level.

***maxChargeDuration attribute*** *(UInt32) [0..1]*

Indicates number of seconds the DER will be able to receive active power at the reserveChargePercent level.

***readingTime attribute*** *(TimeType)*

The timestamp when the DER availability was last updated.

***reserveChargePercent attribute*** *(PerCent) [0..1]*

Percent of continuous received active power (%setMaxChargeRateW) that is estimated to be available in reserve.

***reservePercent attribute*** *(PerCent) [0..1]*

Percent of continuous delivered active power (%setMaxW) that is estimated to be available in reserve.

***statVarAvail attribute*** *(ReactivePower) [0..1]*

Estimated reserve reactive power, in var. Represents the lesser of received or delivered reactive power.

***statWAvail attribute*** *(ActivePower) [0..1]*

Estimated reserve active power, in watts.

**DERCapability Object** (Resource)

Distributed energy resource type and nameplate ratings.

***modesSupported attribute*** *(DERControlType)*

Bitmap indicating the DER Controls implemented by the device. See DERControlType for values.

***rtgAbnormalCategory attribute*** *(UInt8) [0..1]*

Abnormal operating performance category as defined by IEEE 1547-2018. One of:

0 - not specified

1 - Category I

2 - Category II

3 - Category III

All other values reserved.

***rtgMaxA attribute*** *(CurrentRMS) [0..1]*

Maximum continuous AC current capability of the DER, in Amperes (RMS).

***rtgMaxAh attribute*** *(AmpereHour) [0..1]*

Usable energy storage capacity of the DER, in AmpHours.

***rtgMaxChargeRateVA attribute*** *(ApparentPower) [0..1]*

Maximum apparent power charge rating in Volt-Amperes. May differ from the maximum apparent power rating.

***rtgMaxChargeRateW attribute*** *(ActivePower) [0..1]*

Maximum rate of energy transfer received by the storage DER, in Watts.

***rtgMaxDischargeRateVA attribute*** *(ApparentPower) [0..1]*

Maximum apparent power discharge rating in Volt-Amperes. May differ from the maximum apparent power rating.

***rtgMaxDischargeRateW attribute*** *(ActivePower) [0..1]*

Maximum rate of energy transfer delivered by the storage DER, in Watts. Required for combined generation/storage DERs (e.g. DERType == 83).

***rtgMaxV attribute*** *(VoltageRMS) [0..1]*

AC voltage maximum rating.

***rtgMaxVA attribute*** *(ApparentPower) [0..1]*

Maximum continuous apparent power output capability of the DER, in VA.

***rtgMaxVar attribute*** *(ReactivePower) [0..1]*

Maximum continuous reactive power delivered by the DER, in var.

***rtgMaxVarNeg attribute*** *(ReactivePower) [0..1]*

Maximum continuous reactive power received by the DER, in var. If absent, defaults to negative rtgMaxVar.

***rtgMaxW attribute*** *(ActivePower)*

Maximum continuous active power output capability of the DER, in watts. Represents combined generation plus storage output if DERType == 83.

***rtgMaxWh attribute*** *(WattHour) [0..1]*

Maximum energy storage capacity of the DER, in WattHours.

***rtgMinPFOverExcited attribute*** *(PowerFactor) [0..1]*

Minimum Power Factor displacement capability of the DER when injecting reactive power (over-excited); SHALL be a positive value between 0.0 (typically > 0.7) and 1.0. If absent, defaults to unity.

***rtgMinPFUnderExcited attribute*** *(PowerFactor) [0..1]*

Minimum Power Factor displacement capability of the DER when absorbing reactive power (under-excited); SHALL be a positive value between 0.0 (typically > 0.7) and 0.9999. If absent, defaults to rtgMinPFOverExcited.

***rtgMinV attribute*** *(VoltageRMS) [0..1]*

AC voltage minimum rating.

***rtgNormalCategory attribute*** *(UInt8) [0..1]*

Normal operating performance category as defined by IEEE 1547-2018. One of:

0 - not specified

1 - Category A

2 - Category B

All other values reserved.

***rtgOverExcitedPF attribute*** *(PowerFactor) [0..1]*

Specified over-excited power factor.

***rtgOverExcitedW attribute*** *(ActivePower) [0..1]*

Active power rating in Watts at specified over-excited power factor (rtgOverExcitedPF). If present, rtgOverExcitedPF SHALL be present.

***rtgReactiveSusceptance attribute*** *(ReactiveSusceptance) [0..1]*

Reactive susceptance that remains connected to the Area EPS in the cease to energize and trip state.

***rtgUnderExcitedPF attribute*** *(PowerFactor) [0..1]*

Specified under-excited power factor.

***rtgUnderExcitedW attribute*** *(ActivePower) [0..1]*

Active power rating in Watts at specified under-excited power factor (rtgUnderExcitedPF). If present, rtgUnderExcitedPF SHALL be present.

***rtgVNom attribute*** *(VoltageRMS) [0..1]*

AC voltage nominal rating.

***type attribute*** *(DERType)*

Type of DER; see DERType object

**DERControlBase Object** ()

Distributed Energy Resource (DER) control values.

***opModConnect attribute*** *(boolean) [0..1]*

Set DER as connected (true) or disconnected (false). Used in conjunction with ramp rate when re-connecting. Implies galvanic isolation.

***opModEnergize attribute*** *(boolean) [0..1]*

Set DER as energized (true) or de-energized (false). Used in conjunction with ramp rate when re-energizing.

***opModFixedPFAbsorbW attribute*** *(PowerFactorWithExcitation) [0..1]*

The opModFixedPFAbsorbW function specifies a requested fixed Power Factor (PF) setting for when active power is being absorbed. The actual displacement SHALL be within the limits established by setMinPFOverExcited and setMinPFUnderExcited. If issued simultaneously with other reactive power controls (e.g. opModFixedVar) the control resulting in least var magnitude SHOULD take precedence.

***opModFixedPFInjectW attribute*** *(PowerFactorWithExcitation) [0..1]*

The opModFixedPFInjectW function specifies a requested fixed Power Factor (PF) setting for when active power is being injected. The actual displacement SHALL be within the limits established by setMinPFOverExcited and setMinPFUnderExcited. If issued simultaneously with other reactive power controls (e.g. opModFixedVar) the control resulting in least var magnitude SHOULD take precedence.

***opModFixedVar attribute*** *(FixedVar) [0..1]*

The opModFixedVar function specifies the delivered or received reactive power setpoint. The context for the setpoint value is determined by refType and SHALL be one of %setMaxW, %setMaxVar, or %statVarAvail. If issued simultaneously with other reactive power controls (e.g. opModFixedPFInjectW) the control resulting in least var magnitude SHOULD take precedence.

***opModFixedW attribute*** *(SignedPerCent) [0..1]*

The opModFixedW function specifies a requested charge or discharge mode setpoint, in %setMaxChargeRateW if negative value or %setMaxW or %setMaxDischargeRateW if positive value (in hundredths).

***opModFreqDroop attribute*** *(FreqDroopType) [0..1]*

Specifies a frequency-watt operation. This operation limits active power generation or consumption when the line frequency deviates from nominal by a specified amount.

***opModFreqWatt attribute*** *(DERCurveLink) [0..1]*

Specify DERCurveLink for curveType == 0. The Frequency-Watt function limits active power generation or consumption when the line frequency deviates from nominal by a specified amount. The Frequency-Watt curve is specified as an array of Frequency-Watt pairs that are interpolated into a piecewise linear function with hysteresis. The x value of each pair specifies a frequency in Hz. The y value specifies a corresponding active power output in %setMaxW.

***opModHFRTMayTrip attribute*** *(DERCurveLink) [0..1]*

Specify DERCurveLink for curveType == 1. The High Frequency Ride-Through (HFRT) function is specified by one or two duration-frequency curves that define the operating region under high frequency conditions. Each HFRT curve is specified by an array of duration-frequency pairs that will be interpolated into a piecewise linear function that defines an operating region. The x value of each pair specifies a duration (time at a given frequency in seconds). The y value of each pair specifies a frequency, in Hz. This control specifies the "may trip" region.

***opModHFRTMustTrip attribute*** *(DERCurveLink) [0..1]*

Specify DERCurveLink for curveType == 2. The High Frequency Ride-Through (HFRT) function is specified by a duration-frequency curve that defines the operating region under high frequency conditions. Each HFRT curve is specified by an array of duration-frequency pairs that will be interpolated into a piecewise linear function that defines an operating region. The x value of each pair specifies a duration (time at a given frequency in seconds). The y value of each pair specifies a frequency, in Hz. This control specifies the "must trip" region.

***opModHVRTMayTrip attribute*** *(DERCurveLink) [0..1]*

Specify DERCurveLink for curveType == 3. The High Voltage Ride-Through (HVRT) function is specified by one, two, or three duration-volt curves that define the operating region under high voltage conditions. Each HVRT curve is specified by an array of duration-volt pairs that will be interpolated into a piecewise linear function that defines an operating region. The x value of each pair specifies a duration (time at a given voltage in seconds). The y value of each pair specifies an effective percentage voltage, defined as ((locally measured voltage - setVRefOfs / setVRef). This control specifies the "may trip" region.

***opModHVRTMomentaryCessation attribute*** *(DERCurveLink) [0..1]*

Specify DERCurveLink for curveType == 4. The High Voltage Ride-Through (HVRT) function is specified by duration-volt curves that define the operating region under high voltage conditions. Each HVRT curve is specified by an array of duration-volt pairs that will be interpolated into a piecewise linear function that defines an operating region. The x value of each pair specifies a duration (time at a given voltage in seconds). The y value of each pair specifies an effective percent voltage, defined as ((locally measured voltage - setVRefOfs) / setVRef). This control specifies the "momentary cessation" region.

***opModHVRTMustTrip attribute*** *(DERCurveLink) [0..1]*

Specify DERCurveLink for curveType == 5. The High Voltage Ride-Through (HVRT) function is specified by duration-volt curves that define the operating region under high voltage conditions. Each HVRT curve is specified by an array of duration-volt pairs that will be interpolated into a piecewise linear function that defines an operating region. The x value of each pair specifies a duration (time at a given voltage in seconds). The y value of each pair specifies an effective percent voltage, defined as ((locally measured voltage - setVRefOfs) / setVRef). This control specifies the "must trip" region.

***opModLFRTMayTrip attribute*** *(DERCurveLink) [0..1]*

Specify DERCurveLink for curveType == 6. The Low Frequency Ride-Through (LFRT) function is specified by one or two duration-frequency curves that define the operating region under low frequency conditions. Each LFRT curve is specified by an array of duration-frequency pairs that will be interpolated into a piecewise linear function that defines an operating region. The x value of each pair specifies a duration (time at a given frequency in seconds). The y value of each pair specifies a frequency, in Hz. This control specifies the "may trip" region.

***opModLFRTMustTrip attribute*** *(DERCurveLink) [0..1]*

Specify DERCurveLink for curveType == 7. The Low Frequency Ride-Through (LFRT) function is specified by a duration-frequency curve that defines the operating region under low frequency conditions. Each LFRT curve is specified by an array of duration-frequency pairs that will be interpolated into a piecewise linear function that defines an operating region. The x value of each pair specifies a duration (time at a given frequency in seconds). The y value of each pair specifies a frequency, in Hz. This control specifies the "must trip" region.

***opModLVRTMayTrip attribute*** *(DERCurveLink) [0..1]*

Specify DERCurveLink for curveType == 8. The Low Voltage Ride-Through (LVRT) function is specified by one, two, or three duration-volt curves that define the operating region under low voltage conditions. Each LVRT curve is specified by an array of duration-volt pairs that will be interpolated into a piecewise linear function that defines an operating region. The x value of each pair specifies a duration (time at a given voltage in seconds). The y value of each pair specifies an effective percent voltage, defined as ((locally measured voltage - setVRefOfs) / setVRef). This control specifies the "may trip" region.

***opModLVRTMomentaryCessation attribute*** *(DERCurveLink) [0..1]*

Specify DERCurveLink for curveType == 9. The Low Voltage Ride-Through (LVRT) function is specified by duration-volt curves that define the operating region under low voltage conditions. Each LVRT curve is specified by an array of duration-volt pairs that will be interpolated into a piecewise linear function that defines an operating region. The x value of each pair specifies a duration (time at a given voltage in seconds). The y value of each pair specifies an effective percent voltage, defined as ((locally measured voltage - setVRefOfs) / setVRef). This control specifies the "momentary cessation" region.

***opModLVRTMustTrip attribute*** *(DERCurveLink) [0..1]*

Specify DERCurveLink for curveType == 10. The Low Voltage Ride-Through (LVRT) function is specified by duration-volt curves that define the operating region under low voltage conditions. Each LVRT curve is specified by an array of duration-volt pairs that will be interpolated into a piecewise linear function that defines an operating region. The x value of each pair specifies a duration (time at a given voltage in seconds). The y value of each pair specifies an effective percent voltage, defined as ((locally measured voltage - setVRefOfs) / setVRef). This control specifies the "must trip" region.

***opModMaxLimW attribute*** *(PerCent) [0..1]*

The opModMaxLimW function sets the maximum active power generation level at the electrical coupling point as a percentage of set capacity (%setMaxW, in hundredths). This limitation may be met e.g. by reducing PV output or by using excess PV output to charge associated storage.

***opModTargetVar attribute*** *(ReactivePower) [0..1]*

Target reactive power, in var. This control is likely to be more useful for aggregators, as individual DERs may not be able to maintain a target setting.

***opModTargetW attribute*** *(ActivePower) [0..1]*

Target output power, in Watts. This control is likely to be more useful for aggregators, as individual DERs may not be able to maintain a target setting.

***opModVoltVar attribute*** *(DERCurveLink) [0..1]*

Specify DERCurveLink for curveType == 11. The static volt-var function provides over- or under-excited var compensation as a function of measured voltage. The volt-var curve is specified as an array of volt-var pairs that are interpolated into a piecewise linear function with hysteresis. The x value of each pair specifies an effective percent voltage, defined as ((locally measured voltage - setVRefOfs) / setVRef) and SHOULD support a domain of at least 0 - 135. If VRef is present in DERCurve, then the x value of each pair is additionally multiplied by (VRef / 10000). The y value specifies a target var output interpreted as a signed percentage (-100 to 100). The meaning of the y value is determined by yRefType and must be one of %setMaxW, %setMaxVar, or %statVarAvail.

***opModVoltWatt attribute*** *(DERCurveLink) [0..1]*

Specify DERCurveLink for curveType == 12. The Volt-Watt reduces active power output as a function of measured voltage. The Volt-Watt curve is specified as an array of Volt-Watt pairs that are interpolated into a piecewise linear function with hysteresis. The x value of each pair specifies an effective percent voltage, defined as ((locally measured voltage - setVRefOfs) / setVRef) and SHOULD support a domain of at least 0 - 135. The y value specifies an active power output interpreted as a percentage (0 - 100). The meaning of the y value is determined by yRefType and must be one of %setMaxW or %statWAvail.

***opModWattPF attribute*** *(DERCurveLink) [0..1]*

Specify DERCurveLink for curveType == 13. The Watt-PF function varies Power Factor (PF) as a function of delivered active power. The Watt-PF curve is specified as an array of Watt-PF coordinates that are interpolated into a piecewise linear function with hysteresis. The x value of each pair specifies a watt setting in %setMaxW, (0 - 100). The PF output setting is a signed displacement in y value (PF sign SHALL be interpreted according to the EEI convention, where unity PF is considered unsigned). These settings are not expected to be updated very often during the life of the installation, therefore only a single curve is required. If issued simultaneously with other reactive power controls (e.g. opModFixedPFInjectW) the control resulting in least var magnitude SHOULD take precedence.

***opModWattVar attribute*** *(DERCurveLink) [0..1]*

Specify DERCurveLink for curveType == 14. The Watt-Var function varies vars as a function of delivered active power. The Watt-Var curve is specified as an array of Watt-Var pairs that are interpolated into a piecewise linear function with hysteresis. The x value of each pair specifies a watt setting in %setMaxW, (0-100). The y value specifies a target var output interpreted as a signed percentage (-100 to 100). The meaning of the y value is determined by yRefType and must be one of %setMaxW, %setMaxVar, or %statVarAvail.

***rampTms attribute*** *(UInt16) [0..1]*

Requested ramp time, in hundredths of a second, for the device to transition from the current DERControl mode setting(s) to the new mode setting(s). If absent, use default ramp rate (setGradW). Resolution is 1/100 sec.

**DERControl Object** (RandomizableEvent)

Distributed Energy Resource (DER) time/event-based control.

***deviceCategory attribute*** *(DeviceCategoryType) [0..1]*

Specifies the bitmap indicating the categories of devices that SHOULD respond. Devices SHOULD ignore events that do not indicate their device category. If not present, all devices SHOULD respond.

**DERControlList Object** (SubscribableList)

A List element to hold DERControl objects.

**DERControlType Object** (HexBinary32)

Control modes supported by the DER. Bit positions SHALL be defined as follows:

0 - Charge mode

1 - Discharge mode

2 - opModConnect (Connect / Disconnect - implies galvanic isolation)

3 - opModEnergize (Energize / De-Energize)

4 - opModFixedPFAbsorbW (Fixed Power Factor Setpoint when absorbing active power)

5 - opModFixedPFInjectW (Fixed Power Factor Setpoint when injecting active power)

6 - opModFixedVar (Reactive Power Setpoint)

7 - opModFixedW (Charge / Discharge Setpoint)

8 - opModFreqDroop (Frequency-Watt Parameterized Mode)

9 - opModFreqWatt (Frequency-Watt Curve Mode)

10 - opModHFRTMayTrip (High Frequency Ride Through, May Trip Mode)

11 - opModHFRTMustTrip (High Frequency Ride Through, Must Trip Mode)

12 - opModHVRTMayTrip (High Voltage Ride Through, May Trip Mode)

13 - opModHVRTMomentaryCessation (High Voltage Ride Through, Momentary Cessation Mode)

14 - opModHVRTMustTrip (High Voltage Ride Through, Must Trip Mode)

15 - opModLFRTMayTrip (Low Frequency Ride Through, May Trip Mode)

16 - opModLFRTMustTrip (Low Frequency Ride Through, Must Trip Mode)

17 - opModLVRTMayTrip (Low Voltage Ride Through, May Trip Mode)

18 - opModLVRTMomentaryCessation (Low Voltage Ride Through, Momentary Cessation Mode)

19 - opModLVRTMustTrip (Low Voltage Ride Through, Must Trip Mode)

20 - opModMaxLimW (Maximum Active Power)

21 - opModTargetVar (Target Reactive Power)

22 - opModTargetW (Target Active Power)

23 - opModVoltVar (Volt-Var Mode)

24 - opModVoltWatt (Volt-Watt Mode)

25 - opModWattPF (Watt-PowerFactor Mode)

26 - opModWattVar (Watt-Var Mode)

All other values reserved.

**DERCurve Object** (IdentifiedObject)

DER related curves such as Volt-Var mode curves. Relationship between an independent variable (X-axis) and a dependent variable (Y-axis).

***autonomousVRefEnable attribute*** *(boolean) [0..1]*

If the curveType is opModVoltVar, then this field MAY be present. If the curveType is not opModVoltVar, then this field SHALL NOT be present. Enable/disable autonomous vRef adjustment. When enabled, the Volt-Var curve characteristic SHALL be adjusted autonomously as vRef changes and autonomousVRefTimeConstant SHALL be present. If a DER is able to support Volt-Var mode but is unable to support autonomous vRef adjustment, then the DER SHALL execute the curve without autonomous vRef adjustment. If not specified, then the value is false.

***autonomousVRefTimeConstant attribute*** *(UInt32) [0..1]*

If the curveType is opModVoltVar, then this field MAY be present. If the curveType is not opModVoltVar, then this field SHALL NOT be present. Adjustment range for vRef time constant, in hundredths of a second.

***creationTime attribute*** *(TimeType)*

The time at which the object was created.

***curveType attribute*** *(DERCurveType)*

Specifies the associated curve-based control mode.

***openLoopTms attribute*** *(UInt16) [0..1]*

Open loop response time, the time to ramp up to 90% of the new target in response to the change in voltage, in hundredths of a second. Resolution is 1/100 sec. A value of 0 is used to mean no limit. When not present, the device SHOULD follow its default behavior.

***rampDecTms attribute*** *(UInt16) [0..1]*

Decreasing ramp rate, interpreted as a percentage change in output capability limit per second (e.g. %setMaxW / sec). Resolution is in hundredths of a percent/second. A value of 0 means there is no limit. If absent, ramp rate defaults to setGradW.

***rampIncTms attribute*** *(UInt16) [0..1]*

Increasing ramp rate, interpreted as a percentage change in output capability limit per second (e.g. %setMaxW / sec). Resolution is in hundredths of a percent/second. A value of 0 means there is no limit. If absent, ramp rate defaults to rampDecTms.

***rampPT1Tms attribute*** *(UInt16) [0..1]*

The configuration parameter for a low-pass filter, PT1 is a time, in hundredths of a second, in which the filter will settle to 95% of a step change in the input value. Resolution is 1/100 sec.

***vRef attribute*** *(PerCent) [0..1]*

If the curveType is opModVoltVar, then this field MAY be present. If the curveType is not opModVoltVar, then this field SHALL NOT be present. The nominal AC voltage (RMS) adjustment to the voltage curve points for Volt-Var curves.

***xMultiplier attribute*** *(PowerOfTenMultiplierType)*

Exponent for X-axis value.

***yMultiplier attribute*** *(PowerOfTenMultiplierType)*

Exponent for Y-axis value.

***yRefType attribute*** *(DERUnitRefType)*

The Y-axis units context.

**CurrentDERProgramLink Object** (Link)

SHALL contain a Link to an instance of DERProgram. If present, this is the DERProgram containing the currently active DERControl.

**DERCurveList Object** (List)

A List element to hold DERCurve objects.

**CurveData Object** ()

Data point values for defining a curve or schedule

***excitation attribute*** *(boolean) [0..1]*

If yvalue is Power Factor, then this field SHALL be present. If yvalue is not Power Factor, then this field SHALL NOT be present.

True when DER is absorbing reactive power (under-excited), false

when DER is injecting reactive power (over-excited).

***xvalue attribute*** *(Int32)*

The data value of the X-axis (independent) variable, depending on the curve type. See definitions in DERControlBase for further information.

***yvalue attribute*** *(Int32)*

The data value of the Y-axis (dependent) variable, depending on the curve type. See definitions in DERControlBase for further information. If yvalue is Power Factor, the excitation field SHALL be present and yvalue SHALL be a positive value. If yvalue is not Power Factor, the excitation field SHALL NOT be present.

**DERCurveType Object** (UInt8)

0 - opModFreqWatt (Frequency-Watt Curve Mode)

1 - opModHFRTMayTrip (High Frequency Ride Through, May Trip Mode)

2 - opModHFRTMustTrip (High Frequency Ride Through, Must Trip Mode)

3 - opModHVRTMayTrip (High Voltage Ride Through, May Trip Mode)

4 - opModHVRTMomentaryCessation (High Voltage Ride Through, Momentary Cessation Mode)

5 - opModHVRTMustTrip (High Voltage Ride Through, Must Trip Mode)

6 - opModLFRTMayTrip (Low Frequency Ride Through, May Trip Mode)

7 - opModLFRTMustTrip (Low Frequency Ride Through, Must Trip Mode)

8 - opModLVRTMayTrip (Low Voltage Ride Through, May Trip Mode)

9 - opModLVRTMomentaryCessation (Low Voltage Ride Through, Momentary Cessation Mode)

10 - opModLVRTMustTrip (Low Voltage Ride Through, Must Trip Mode)

11 - opModVoltVar (Volt-Var Mode)

12 - opModVoltWatt (Volt-Watt Mode)

13 - opModWattPF (Watt-PowerFactor Mode)

14 - opModWattVar (Watt-Var Mode)

All other values reserved.

**DERProgram Object** (SubscribableIdentifiedObject)

Distributed Energy Resource program.

***primacy attribute*** *(PrimacyType)*

Indicates the relative primacy of the provider of this Program.

**DERProgramList Object** (SubscribableList)

A List element to hold DERProgram objects.

***pollRate attribute*** *(UInt32) [0..1] «XSDattribute»*

The default polling rate for this function set (this resource and all resources below), in seconds. If not specified, a default of 900 seconds (15 minutes) is used. It is RECOMMENDED a client poll the resources of this function set every pollRate seconds.

**DERStatus Object** (SubscribableResource)

DER status information.

***alarmStatus attribute*** *(HexBinary32) [0..1]*

Bitmap indicating the status of DER alarms (see DER LogEvents for more details).

0 - DER\_FAULT\_OVER\_CURRENT

1 - DER\_FAULT\_OVER\_VOLTAGE

2 - DER\_FAULT\_UNDER\_VOLTAGE

3 - DER\_FAULT\_OVER\_FREQUENCY

4 - DER\_FAULT\_UNDER\_FREQUENCY

5 - DER\_FAULT\_VOLTAGE\_IMBALANCE

6 - DER\_FAULT\_CURRENT\_IMBALANCE

7 - DER\_FAULT\_EMERGENCY\_LOCAL

8 - DER\_FAULT\_EMERGENCY\_REMOTE

9 - DER\_FAULT\_LOW\_POWER\_INPUT

10 - DER\_FAULT\_PHASE\_ROTATION

11-31 - Reserved

***genConnectStatus attribute*** *(ConnectStatusType) [0..1]*

Connect/status value for generator DER.

See ConnectStatusType for values.

***inverterStatus attribute*** *(InverterStatusType) [0..1]*

DER InverterStatus/value.

See InverterStatusType for values.

***localControlModeStatus attribute*** *(LocalControlModeStatusType) [0..1]*

The local control mode status.

See LocalControlModeStatusType for values.

***manufacturerStatus attribute*** *(ManufacturerStatusType) [0..1]*

Manufacturer status code.

***operationalModeStatus attribute*** *(OperationalModeStatusType) [0..1]*

Operational mode currently in use.

See OperationalModeStatusType for values.

***readingTime attribute*** *(TimeType)*

The timestamp when the current status was last updated.

***stateOfChargeStatus attribute*** *(StateOfChargeStatusType) [0..1]*

State of charge status.

See StateOfChargeStatusType for values.

***storageModeStatus attribute*** *(StorageModeStatusType) [0..1]*

Storage mode status.

See StorageModeStatusType for values.

***storConnectStatus attribute*** *(ConnectStatusType) [0..1]*

Connect/status value for storage DER.

See ConnectStatusType for values.

**DERUnitRefType Object** (UInt8)

Specifies context for interpreting percent values:

0 - N/A

1 - %setMaxW

2 - %setMaxVar

3 - %statVarAvail

4 - %setEffectiveV

5 - %setMaxChargeRateW

6 - %setMaxDischargeRateW

7 - %statWAvail

All other values reserved.

**CurrentRMS Object** ()

Average flow of charge through a conductor.

***multiplier attribute*** *(PowerOfTenMultiplierType)*

Specifies exponent of value.

***value attribute*** *(UInt16)*

Value in amperes RMS (uom 5)

**FixedPointType Object** ()

Abstract type for specifying a fixed-point value without a given unit of measure.

***multiplier attribute*** *(PowerOfTenMultiplierType)*

Specifies exponent of uom.

***value attribute*** *(Int16)*

Dimensionless value

**UnsignedFixedPointType Object** ()

Abstract type for specifying an unsigned fixed-point value without a given unit of measure.

***multiplier attribute*** *(PowerOfTenMultiplierType)*

Specifies exponent of uom.

***value attribute*** *(UInt16)*

Dimensionless value

**ActivePower Object** ()

The active (real) power P (in W) is the product of root-mean-square (RMS) voltage, RMS current, and cos(theta) where theta is the phase angle of current relative to voltage. It is the primary measure of the rate of flow of energy.

***multiplier attribute*** *(PowerOfTenMultiplierType)*

Specifies exponent for uom.

***value attribute*** *(Int16)*

Value in watts (uom 38)

**AmpereHour Object** ()

Available electric charge

***multiplier attribute*** *(PowerOfTenMultiplierType)*

Specifies exponent of uom.

***value attribute*** *(UInt16)*

Value in ampere-hours (uom 106)

**ApparentPower Object** ()

The apparent power S (in VA) is the product of root mean square (RMS) voltage and RMS current.

***multiplier attribute*** *(PowerOfTenMultiplierType)*

Specifies exponent of uom.

***value attribute*** *(UInt16)*

Value in volt-amperes (uom 61)

**ReactivePower Object** ()

The reactive power Q (in var) is the product of root mean square (RMS) voltage, RMS current, and sin(theta) where theta is the phase angle of current relative to voltage.

***multiplier attribute*** *(PowerOfTenMultiplierType)*

Specifies exponent of uom.

***value attribute*** *(Int16)*

Value in volt-amperes reactive (var) (uom 63)

**ReactiveSusceptance Object** ()

Reactive susceptance

***multiplier attribute*** *(PowerOfTenMultiplierType)*

Specifies exponent of uom.

***value attribute*** *(UInt16)*

Value in siemens (uom 53)

**PowerFactor Object** ()

Specifies a setpoint for Displacement Power Factor, the ratio between apparent and active powers at the fundamental frequency (e.g. 60 Hz).

***displacement attribute*** *(UInt16)*

Significand of an unsigned value of cos(theta) between 0 and 1.0. E.g. a value of 0.95 may be specified as a displacement of 950 and a multiplier of -3.

***multiplier attribute*** *(PowerOfTenMultiplierType)*

Specifies exponent of 'displacement'.

**PowerFactorWithExcitation Object** ()

Specifies a setpoint for Displacement Power Factor, the ratio between apparent and active powers at the fundamental frequency (e.g. 60 Hz) and includes an excitation flag.

***displacement attribute*** *(UInt16)*

Significand of an unsigned value of cos(theta) between 0 and 1.0. E.g. a value of 0.95 may be specified as a displacement of 950 and a multiplier of -3.

***excitation attribute*** *(boolean)*

True when DER is absorbing reactive power (under-excited), false

when DER is injecting reactive power (over-excited).

***multiplier attribute*** *(PowerOfTenMultiplierType)*

Specifies exponent of 'displacement'.

**FixedVar Object** ()

Specifies a signed setpoint for reactive power.

***refType attribute*** *(DERUnitRefType)*

Indicates whether to interpret 'value' as %setMaxVar or %statVarAvail.

***value attribute*** *(SignedPerCent)*

Specify a signed setpoint for reactive power in % (see 'refType' for context).

**WattHour Object** ()

Active (real) energy

***multiplier attribute*** *(PowerOfTenMultiplierType)*

Specifies exponent of uom.

***value attribute*** *(UInt16)*

Value in watt-hours (uom 72)

**VoltageRMS Object** ()

Average electric potential difference between two points.

***multiplier attribute*** *(PowerOfTenMultiplierType)*

Specifies exponent of uom.

***value attribute*** *(UInt16)*

Value in volts RMS (uom 29)

**ConnectStatusType Object** ()

DER ConnectStatus value (bitmap):

0 - Connected

1 - Available

2 - Operating

3 - Test

4 - Fault / Error

All other values reserved.

***dateTime attribute*** *(TimeType)*

The date and time at which the state applied.

***value attribute*** *(HexBinary8)*

The value indicating the state.

**InverterStatusType Object** ()

DER InverterStatus value:

0 - N/A

1 - off

2 - sleeping (auto-shutdown) or DER is at low output power/voltage

3 - starting up or ON but not producing power

4 - tracking MPPT power point

5 - forced power reduction/derating

6 - shutting down

7 - one or more faults exist

8 - standby (service on unit) - DER may be at high output voltage/power

9 - test mode

10 - as defined in manufacturer status

All other values reserved.

***dateTime attribute*** *(TimeType)*

The date and time at which the state applied.

***value attribute*** *(UInt8)*

The value indicating the state.

**LocalControlModeStatusType Object** ()

DER LocalControlModeStatus/value:

0 – local control

1 – remote control

All other values reserved.

***dateTime attribute*** *(TimeType)*

The date and time at which the state applied.

***value attribute*** *(UInt8)*

The value indicating the state.

**ManufacturerStatusType Object** ()

DER ManufacturerStatus/value: String data type

***dateTime attribute*** *(TimeType)*

The date and time at which the state applied.

***value attribute*** *(String6)*

The value indicating the state.

**OperationalModeStatusType Object** ()

DER OperationalModeStatus value:

0 - Not applicable / Unknown

1 - Off

2 - Operational mode

3 - Test mode

All other values reserved.

***dateTime attribute*** *(TimeType)*

The date and time at which the state applied.

***value attribute*** *(UInt8)*

The value indicating the state.

**StateOfChargeStatusType Object** ()

DER StateOfChargeStatus value: Percent data type

***dateTime attribute*** *(TimeType)*

The date and time at which the state applied.

***value attribute*** *(PerCent)*

The value indicating the state.

**StorageModeStatusType Object** ()

DER StorageModeStatus value:

0 – storage charging

1 – storage discharging

2 – storage holding

All other values reserved.

***dateTime attribute*** *(TimeType)*

The date and time at which the state applied.

***value attribute*** *(UInt8)*

The value indicating the state.

### **Links Package**

Contains definitions of Link specializations used to require certain associations.

**AccountBalanceLink Object** (Link)

SHALL contain a Link to an instance of AccountBalance.

**ActiveBillingPeriodListLink Object** (ListLink)

SHALL contain a Link to a List of active BillingPeriod instances.

**ActiveCreditRegisterListLink Object** (ListLink)

SHALL contain a Link to a List of active CreditRegister instances.

**ActiveDERControlListLink Object** (ListLink)

SHALL contain a Link to a List of active DERControl instances.

**ActiveEndDeviceControlListLink Object** (ListLink)

SHALL contain a Link to a List of active EndDeviceControl instances.

**ActiveFlowReservationListLink Object** (ListLink)

SHALL contain a Link to a List of active FlowReservation instances.

**ActiveProjectionReadingListLink Object** (ListLink)

SHALL contain a Link to a List of active ProjectionReading instances.

**ActiveSupplyInterruptionOverrideListLink Object** (ListLink)

SHALL contain a Link to a List of active SupplyInterruptionOverride instances.

**ActiveTargetReadingListLink Object** (ListLink)

SHALL contain a Link to a List of active TargetReading instances.

**ActiveTextMessageListLink Object** (ListLink)

SHALL contain a Link to a List of active TextMessage instances.

**ActiveTimeTariffIntervalListLink Object** (ListLink)

SHALL contain a Link to a List of active TimeTariffInterval instances.

**AssociatedDERProgramListLink Object** (ListLink)

SHALL contain a Link to a List of DERPrograms having the DERControl(s) for this DER.

**AssociatedUsagePointLink Object** (Link)

SHALL contain a Link to an instance of UsagePoint. If present, this is the submeter that monitors the DER output.

**BillingPeriodListLink Object** (ListLink)

SHALL contain a Link to a List of BillingPeriod instances.

**BillingReadingListLink Object** (ListLink)

SHALL contain a Link to a List of BillingReading instances.

**BillingReadingSetListLink Object** (ListLink)

SHALL contain a Link to a List of BillingReadingSet instances.

**ConfigurationLink Object** (Link)

SHALL contain a Link to an instance of Configuration.

**ConsumptionTariffIntervalListLink Object** (ListLink)

SHALL contain a Link to a List of ConsumptionTariffInterval instances.

**CreditRegisterListLink Object** (ListLink)

SHALL contain a Link to a List of CreditRegister instances.

**CustomerAccountLink Object** (Link)

SHALL contain a Link to an instance of CustomerAccount.

**CustomerAccountListLink Object** (ListLink)

SHALL contain a Link to a List of CustomerAccount instances.

**CustomerAgreementListLink Object** (ListLink)

SHALL contain a Link to a List of CustomerAgreement instances.

**DemandResponseProgramLink Object** (Link)

SHALL contain a Link to an instance of DemandResponseProgram.

**DemandResponseProgramListLink Object** (ListLink)

SHALL contain a Link to a List of DemandResponseProgram instances.

**DERAvailabilityLink Object** (Link)

SHALL contain a Link to an instance of DERAvailability.

**DERCapabilityLink Object** (Link)

SHALL contain a Link to an instance of DERCapability.

**DefaultDERControlLink Object** (Link)

SHALL contain a Link to an instance of DefaultDERControl. This is the default mode of the DER which MAY be overridden by DERControl events.

**DERControlListLink Object** (ListLink)

SHALL contain a Link to a List of DERControl instances.

**DERCurveLink Object** (Link)

SHALL contain a Link to an instance of DERCurve.

**DERCurveListLink Object** (ListLink)

SHALL contain a Link to a List of DERCurve instances.

**DERLink Object** (Link)

SHALL contain a Link to an instance of DER.

**DERListLink Object** (ListLink)

SHALL contain a Link to a List of DER instances.

**DERProgramLink Object** (Link)

SHALL contain a Link to an instance of DERProgram.

**DERProgramListLink Object** (ListLink)

SHALL contain a Link to a List of DERProgram instances.

**DERSettingsLink Object** (Link)

SHALL contain a Link to an instance of DERSettings.

**DERStatusLink Object** (Link)

SHALL contain a Link to an instance of DERStatus.

**DeviceCapabilityLink Object** (Link)

SHALL contain a Link to an instance of DeviceCapability.

**DeviceInformationLink Object** (Link)

SHALL contain a Link to an instance of DeviceInformation.

**DeviceStatusLink Object** (Link)

SHALL contain a Link to an instance of DeviceStatus.

**EndDeviceControlListLink Object** (ListLink)

SHALL contain a Link to a List of EndDeviceControl instances.

**EndDeviceLink Object** (Link)

SHALL contain a Link to an instance of EndDevice.

**EndDeviceListLink Object** (ListLink)

SHALL contain a Link to a List of EndDevice instances.

**FileLink Object** (Link)

This element MUST be set to the URI of the most recent File being loaded/activated by the LD. In the case of file status 0, this element MUST be omitted.

**FileListLink Object** (ListLink)

SHALL contain a Link to a List of File instances.

**FileStatusLink Object** (Link)

SHALL contain a Link to an instance of FileStatus.

**FlowReservationRequestListLink Object** (ListLink)

SHALL contain a Link to a List of FlowReservationRequest instances.

**FlowReservationResponseListLink Object** (ListLink)

SHALL contain a Link to a List of FlowReservationResponse instances.

**FunctionSetAssignmentsListLink Object** (ListLink)

SHALL contain a Link to a List of FunctionSetAssignments instances.

**HistoricalReadingListLink Object** (ListLink)

SHALL contain a Link to a List of HistoricalReading instances.

**IPAddrListLink Object** (ListLink)

SHALL contain a Link to a List of IPAddr instances.

**IPInterfaceListLink Object** (ListLink)

SHALL contain a Link to a List of IPInterface instances.

**LLInterfaceListLink Object** (ListLink)

SHALL contain a Link to a List of LLInterface instances.

**LoadShedAvailabilityListLink Object** (ListLink)

SHALL contain a Link to a List of LoadShedAvailability instances.

**LogEventListLink Object** (ListLink)

SHALL contain a Link to a List of LogEvent instances.

**MessagingProgramListLink Object** (ListLink)

SHALL contain a Link to a List of MessagingProgram instances.

**MeterReadingLink Object** (Link)

SHALL contain a Link to an instance of MeterReading.

**MeterReadingListLink Object** (ListLink)

SHALL contain a Link to a List of MeterReading instances.

**MirrorUsagePointListLink Object** (ListLink)

SHALL contain a Link to a List of MirrorUsagePoint instances.

**NeighborListLink Object** (ListLink)

SHALL contain a Link to a List of Neighbor instances.

**NotificationListLink Object** (ListLink)

SHALL contain a Link to a List of Notification instances.

**PowerStatusLink Object** (Link)

SHALL contain a Link to an instance of PowerStatus.

**PrepaymentLink Object** (Link)

SHALL contain a Link to an instance of Prepayment.

**PrepaymentListLink Object** (ListLink)

SHALL contain a Link to a List of Prepayment instances.

**PrepayOperationStatusLink Object** (Link)

SHALL contain a Link to an instance of PrepayOperationStatus.

**PriceResponseCfgListLink Object** (ListLink)

SHALL contain a Link to a List of PriceResponseCfg instances.

**ProjectionReadingListLink Object** (ListLink)

SHALL contain a Link to a List of ProjectionReading instances.

**RateComponentLink Object** (Link)

SHALL contain a Link to an instance of RateComponent.

**RateComponentListLink Object** (ListLink)

SHALL contain a Link to a List of RateComponent instances.

**ReadingLink Object** (Link)

A Link to a Reading.

**ReadingListLink Object** (ListLink)

SHALL contain a Link to a List of Reading instances.

**ReadingSetListLink Object** (ListLink)

SHALL contain a Link to a List of ReadingSet instances.

**ReadingTypeLink Object** (Link)

SHALL contain a Link to an instance of ReadingType.

**RegistrationLink Object** (Link)

SHALL contain a Link to an instance of Registration.

**ResponseListLink Object** (ListLink)

SHALL contain a Link to a List of Response instances.

**ResponseSetListLink Object** (ListLink)

SHALL contain a Link to a List of ResponseSet instances.

**RPLInstanceListLink Object** (ListLink)

SHALL contain a Link to a List of RPLInterface instances.

**RPLSourceRoutesListLink Object** (ListLink)

SHALL contain a Link to a List of RPLSourceRoutes instances.

**SelfDeviceLink Object** (Link)

SHALL contain a Link to an instance of SelfDevice.

**ServiceSupplierLink Object** (Link)

SHALL contain a Link to an instance of ServiceSupplier.

**SubscriptionListLink Object** (ListLink)

SHALL contain a Link to a List of Subscription instances.

**SupplyInterruptionOverrideListLink Object** (ListLink)

SHALL contain a Link to a List of SupplyInterruptionOverride instances.

**SupportedLocaleListLink Object** (ListLink)

SHALL contain a Link to a List of SupportedLocale instances.

**TargetReadingListLink Object** (ListLink)

SHALL contain a Link to a List of TargetReading instances.

**TariffProfileLink Object** (Link)

SHALL contain a Link to an instance of TariffProfile.

**TariffProfileListLink Object** (ListLink)

SHALL contain a Link to a List of TariffProfile instances.

**TextMessageListLink Object** (ListLink)

SHALL contain a Link to a List of TextMessage instances.

**TimeLink Object** (Link)

SHALL contain a Link to an instance of Time.

**TimeTariffIntervalListLink Object** (ListLink)

SHALL contain a Link to a List of TimeTariffInterval instances.

**UsagePointLink Object** (Link)

SHALL contain a Link to an instance of UsagePoint.

**UsagePointListLink Object** (ListLink)

SHALL contain a Link to a List of UsagePoint instances.