|  |
| --- |
|  |

|  |
| --- |
| CHANGE REQUEST |
| Meeting ID:\* | RDM 44 |
| Source:\* | Bob Flynn, Convida Wireless , Bob.Flynn@convidawireless.comCyrille Bareau, Orange, cyrille.bareau@orange.comLeila Le Brun, Orange, leila.lebrun@orange.com Marianne Mohali, Orange, marianne.mohali@orange.com Przemyslaw Ratuszek, Orange, przemyslaw.ratuszek@orange.com Andreas Kraft, Deutsche Telekom, Andreas.Kraft@t-systems.com |
| Date:\* | 2020-02-09 |
| Reason for Change/s:\* | Clarification of table parameters |
| CR against: Release\* | Rel-4 |
| CR against: WI\* | [ ]  Active < > [x]  MNT maintenance / < Work Item number(optional)>Is this a mirror CR? Yes [ ]  No [ ] mirror CR number: (Note to Rapporteur - use latest agreed revision)[ ]  STE Small Technical Enhancements / < Work Item number (optional)>Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0023 v4.3.0 |
| Clauses \* | 5.2.2 |
| Type of change: \* | [ ]  Editorial change[ ]  Bug Fix or Correction[x]  Change to existing feature or functionality[ ]  New feature or functionalityOnly ONE of the above shall be ticked |
| Other TS/TR(s) impacted | None |
| Post Freeze checking:\* | This CR contains only essential changes and corrections? YES [x]  NO [ ] This CR may break backwards compatibility with the last approved version of the TS? YES [ ]  NO [ ]  |
| Template Version: January 2019 (do not modify) |

**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

GUIDELINES for Change Requests:

Provide an informative introduction containing the problem(s) being solved, and a summary list of proposals.

Each CR should contain changes related to only one particular issue/problem.

In case of a correction, and the change apply to previous releases, a separate “mirror CR” should be posted at the same time of this CR

Mirror CR: applies only when the text, including clause numbering are exactly the same.

Companion CR: applies when the change means the same but the baselines differ in some way (e.g. clause number).

Follow the principle of completeness, where all changes related to the issue or problem within a deliverable are simultaneously proposed to be made E.g. A change impacting 5 tables should not only include a proposal to change only 3 tables. Includes any changes to references, definitions, and acronyms in the same deliverable.

Follow the drafting rules.

All pictures must be editable.

Check spelling and grammar to the extent practicable.

Use Change bars for modifications.

The change should include the current and surrounding clauses to clearly show where a change is located and to provide technical context of the proposed change. Additions of complete clauses need not show surrounding clauses as long as the proposed clause number clearly shows where the new clause is proposed to be located.

Multiple changes in a single CR shall be clearly separated by horizontal lines with embedded text such as, start of change 1, end of change 1, start of new clause, end of new clause.

When subsequent changes are made to content of a CR, then the accepted version should not show changes over changes. The accepted version of the CR should only show changes relative to the baseline approved text.

## Introduction

In TS-0023 the description of a module’s datapoints includes a R/W column that contains values R or RW. There may be “W” but I did not check the entire document as it does not impact the purpose of the contribution.

For example refer to *Table 5.3.1.1-2: DataPoints of 3Dprinter ModuleClass*

In TS-0004 a similar column exist to describe the parameter presence in a Create or Update request. Refer to *Table 7.4.1.1-2.*

These two concepts are not intended to be related to each other.

This contribution serves to explain the use in TS-0023 for better clarity and interoperability.

## ---------------------- Start of Change 1--------------------------

### Description rules for Module Classes and Device models

When the Home Appliances Information Model is described based on SDT, the following rules shall be applied:

* Rule 1: CamelCase rule:
* When naming each element, lowerCamelCase shall be used as the Java coding rules [2].
* Rule 2: Rule for description of Action, DataPoint:
* DataPoint shall be used to represent stateless operations. (e.g. powerState of binarySwitch for on/off operations).
* Action shall be used when describing stateful condition, handling unknown internal state conditions (e.g. upVolume/downVolume by increasing/decreasing the audioVolume in steps, handling transactional procedures, or checking integrity using username plus password at the same time).
* Rule 3: Rule for description of DataPoint and Property:
* Non-functional information shall be described as a Property. Functional information shall be described as a DataPoint. (E.g. non-functional information: version, id; functional information: targetTemperature, targetVolume).
* Rule 4: Definition of the Domain:
* The Domain, in the case of the Home Appliance Information Model, is specified as "org.onem2m.home".
* The sub-domain for Device and ModuleClass shall be specified as "org.onem2m.home.devices" and "org.onem2m.home.moduleclasses" respectively.
* Rule 5: Naming rule for the element:
* The name of each element should be concise and avoid repeating its parent element name; but
* It may include the name of its parent element for readability. (e.g., lightDimmerUp, lightDimmerDown under lightDimmer).
* Rule 6: Criteria for marking elements as optional or mandatory:
* An element shall only be defined as mandatory if it's foreseen to be universally mandatory to all implementing technologies.
* Rule 7: Enumeration type:
* When describing the meaning of values for enumeration type elements, they may be described in another clause.
* The enumeration types for Home Appliance Information Model are based on <xs:integer>, and the numeric values are interpreted as specified in clause 5.6.
* The name of an enumeration type shall start with the prefix “enum”. This prefix shall not be used with non-enumaration type names.
* Rule 8: Rule for unit in documentation :
* SI (International Systems of Units in [z]) measurement (e.g. meter, kilogram, second.) should be considered as first candidate.
* Otherwise, it may be kept consistency with implementing technologies such as other SDO’s specification..
* Units of measures shall be given in the form of a shortcut compliant to table 5.2.1-1.

Table 5.2.1-1: Shortcuts for units

|  |  |  |
| --- | --- | --- |
| Original name | Short name | Explanation |
| Ampere | A |  |
| Ampere Hour | Ah |  |
| Bar | bar |  |
| Celsius | °C |  |
| Centimeters | cm |  |
| Cubic Meter | m3 |  |
| Cubic meter per hour | m3/h |  |
| Decibel | dB |  |
| Decibel-milliwatts | dBm |  |
| Degrees | deg |  |
| g-force | g-f |  |
| Grams | g |  |
| Hertz | Hz |  |
| Kilocalories | kcal |  |
| Kilocalories per hour | kcal/h |  |
| Kilograms per square meter | kg/m2 |  |
| Kilopascal | kPa |  |
| kilovar | kvar |  |
| Kilowatt | kW |  |
| Megabyte | MB | 1 MB = 1024 \* 1024 bytes |
| MegaHertz | MHz |  |
| Meter | m |  |
| Meters per second | m/s |  |
| Miligram per cubic meter | mg/m3 |  |
| Microgram per cubic meter | μg/m3 |  |
| Milligram per deciliter | mg/dl |  |
| Milligram per liter | mg/L |  |
| Millimeter | mm |  |
| Millimeter of mercury | mmHg |  |
| Milliseconds | ms |  |
| Milliwatt per cubic centimetre | mW/cm2 |  |
| Minute | min |  |
| Odor unit per cubic meter | OU/m3 |  |
| Ohm | ohm |  |
| Parts per minute | ppm |  |
| Percent | pct |  |
| Picofarad | pF |  |
| Seconds | s  |  |
| Siemens per meter | S/m |  |
| Volt | V |  |
| Watt | W |  |
| Watt hour | Wh |  |

*

Editor's note: Popular unit in particular industrial domain shall be considered (e.g. cm for human height, calories for energy consumption in healthcare domain). It shall be made coherent in the document, as possible.

* Rule 9: Rule for type :
* Measured and/or calculated values should be represented in float (without taking care of resolution of values).

Editor’s note: It should be made coherent in the document, as possible. Unit shall not be fixed as a rule but be decided with correspondence to each device or module.

* Rule 10: Inheritance of ModuleClasses :
* A ModuleClass may inherit from another existing ModuleClass in order to provide additional functionalities based on the existing ModuleClass. However, inheritance from multiple ModuleClasses is not allowed (due to the “diamond problem” [i.6]).
* Inheritance of ModuleClass shall only be used in the case that extending an existing ModuleClass is not appropriate, i.e. the functionality to be added is irrelevant to the original design purpose of the existing ModuleClass (e.g. adding a ‘time’ DataPoint to a ‘binarySwitch’ ModuleClass).
* Rule 11: When to differentiate between current and target Data Points in ModuleClasses:
* Device operations, which are executed when setting data points to specific values, may take some time to reach the desired result. For example, setting a new temperature to a heater does not immediately change the room temperature, but it may take some time for the heater to increase the temperature. Therefore, it is sometimes necessary to distinguish between current and target data points.
* A ModuleClass must provide an additional “target” data point when the “current” data point …
	+ is writable, and
	+ the functionality that is mapped to the data point is an operation, not a configuration function, and
	+ the operation may take some time to start and/or to complete, or reach the desired result.
* When a ModuleClass provides current and target data points then the name for the current data point must have the prefix “current”, and the name for the target data point must have the prefix “target”. Both data points must have the same suffix, for example “currentTemperature” and “targetTemperature”.
* Rule 12: Algorithm to generate short names for Devices, ModuleClasses, Data Points, Actions
* Every domain in oneM2M defines their own short names, i.e. there may exist the same short name in more than one domain, but these short names are distinguished by the domain prefix.
* Previous defined short names of the home domain, e.g. from a previous version of the specification, must be taken into account. They are assigned to the same original names.
* The algorithm to generate the short names from the original names works as follows:
	+ The maximum length of a short name for TS-0023 is 5 characters. This length includes the optional appended distinguishing number (see below), but not the suffix for announced resources.
	+ If the length of the original name is equal or less than 5 characters, then store the original name as an intermediate result.
	+ Else, if the length of the original name is greater than 5 characters, then perform the following procedure:
		- The first and the last character of the original name are stored as first and second character as an intermediate result.
		- All the upper-case characters of the original name, starting with the first upper-case character, are inserted one by one before the last character of the intermediate result, up to a total length of 5 characters of the intermediate result.
		- In case the length of the intermediate result after these steps is less than 5 characters, then the intermediate result is filled with characters from the original string until the length of the intermediate result is 5 characters, following this procedure: the second character of the original name is inserted as the second character of the intermediate result while shifting all characters from the intermediate result by one character forward. This is repeated with the third, fourth, etc., character from the original name.
	+ The intermediate result is now compared with all existing short names. If the intermediate result can be found in the list of existing short names, then execute the following steps until the intermediate result cannot be found in the list of previously defined short names:
		- Replace the last character of the intermediate result with an integer number, starting with 0. If the number becomes a two-digit number, then replace the last two characters of the intermediate result, and so forth.
		- Repeat the check described above. If the intermediate result is still the same as an existing short name, then the appended integer number is increased by 1, and the check is repeated.
	+ The intermediate result is now stored as a new short name in the list of existing short names.
* Short names for announced resources are created by taking the regular short name of the entity and appending the characters “Annc” to it. Short names for announced resources therefore have a maximum length of 9 characters.

### The following table provides some examples for short names that have been created by the described algorithm.

Table 5.2.1-2: Examples for original name to short name mappings

|  |  |
| --- | --- |
| Original name | short name |
| co2 | co2 |
| clock | clock |
| currentJobMode | cuJMe |
| absoluteStartTime | abSTe |
| absoluteStopTime | abST0 |
| impactSensor | impSr |
| impactSensorAnnc | impSrAnnc |

• Rule 13: Rule for R/W column

* The value used in this column defines the interface as it applies to the user of this module. The entity that this module represents (device AE or IPE AE) can read or write to any or all of the datapoints as needed in order to implement the defined interface to the user. <accessControlPolicy> resources shall be defined to enforce access control to the datapoints of the module defined such that R in the R/W column has READ *accessControlOperations* and RW in the R/W column has READ and UPDATE *accessControlOperations*.

-------------------------------------------------- End of Change 1---------------------------------------------------