|  |
| --- |
|  |

|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* | SDS 40 |
| Source:\* | Bob Flynn, Convida Wireless , Bob.Flynn@convidawireless.com |
| Date:\* | 2019-05-14 |
| Reason for Change/s:\* | New attribute for <timeSeries> |
| CR against: Release\* | Rel-3 |
| CR against: WI\* | Active <Work Item number>  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-0001 v3.15.0 |
| Clauses \* | 9.6.36 |
| Type of change: \* | Editorial change  Bug Fix or Correction  Change to existing feature or functionality  New feature or functionality  Only 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  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

This contribution addresses:

* New attribute *periodicIntervalDelta* for <timeSeries>
* Additional condition for *missingDataDetectTimer*

In certain scenarios, applications might not generate time-series data at the exact desired time. Applications might generate time-series data ‘x’ millisec lesser or greater relative to *dataGenerationTime*.

As it would be very stringent to expect applications to generate time-series data at the exact millsecond, proposal is to add a new attribute *periodicIntervalDelta* , the value of which will define the delta time for *periodicInterval.*

With this, data generation time for time-series data will be considered for an interval (*periodicInterval* +/- *periodicIntervalDelta*).

A limit must be defined on *periodicIntervalDelta* value otherwise it will create a conflict for *dataGenerationTime* of consecutive <*timeSeriesInstance*>. The value of *periodicIntervalDelta* must be lesser than (*periodicInterval/2*).

Additionally, *missingDataDetectTimer* must be greater than *periodicIntervalDelta* if *periodicIntervalDelta* is present. This is to ensure that time-series data is not declared as missed even before *periodicIntervalDelta* is considered.

R01 – During offline review, the question about drift of a <timeSeriesInstance) was considered.

For the purpose of computing the next expected *dataGenerationTime* should that be computed based on the “current expected data generation time” or should is be computed based on the “current received data generation time”?

For example: if dgt = 13:00 and periodicInterval is 1 hour and *periodIntervalDelta* is 2 minutes.

Currently we would expect arrival of data at 14:00, 15:00, 16:00,…

The detection would allow a +/- 5 minute value for those dgt values. If 14:03 arrives, data is not missed (next window is 14:55 – 15:05). If 14:08 arrives, data is missed. If 14:55 arrives, data is not missed. If 15:08 arrives, data is missed.

Alternatively, if we account for drift the following would occur: If 14:03 arrives, data is not missed(next window is 14:57 – 15:08). If 14:08 arrives, data is missed. If 14:55 arrives, data is missed. If 15:08 arrives, data is not missed.

Current specification supports the first described approach – the question raised is should we support the second approach?

R02 – Review notes that I received

- should we only do this in Rel-4 and not Rel-3.

*I believe that the periodicIntervalDelta is needed in Rel-3 because we made this a feature to be tested for compliance in Rel-3. I think this change is important from a practical implementation perspective. However, if the majority thinks only Rel-4, the impact on the certification tests is minimal, it can be handled with some difficulty. My push back would be, why not Rel-3*.

- Regarding drift, additional clarification is needed to explain whether periodic timer is anchored to dgt of first timeSeriesInstance or gets recalibrated to the dgt of a new timeSeriesInstance that is successfully received/processed.

*The Drift discussion is only discussion. The current specification does not support the drift concept and the periodic timer is anchored to the dgt of the first <tsi>. The question that I am raising regarding whether we should support drift would in fact recalibrate to the dgt of a new <tsi> that is successfully received – if we decide to support drift. This is something that can be Rel-4, as this is new functionality. I think that this would be a separate contribution because this is not “fixing or completing” current functionality but rather adding a new functionality. If we decide we want to support the drift concept, I can be given an action to come back with Rel-4 contribution.*

- the definition of the missingDataDetectTimer in TS-0001 is confusing and needs further clarification since it is not obvious how this timer works in conjunction with periodicInterval timer.  Also, the TCP part should probably be deleted.

"The *missingDataDetectTimer* after which a missing Time Series Data shall be considered lost by the hosting CSE. Note that the setting of this value may not apply in certain transports such as TCP, and as such the hosting CSE may reject proposed values or suggest different values."

Changes made below

### -------------------------------------Start of change 1-------------------------------------------

Table 9.6.36-2: Attributes of <*timeSeries*> resource

| Attributes of  *<timeSeries>* | Multiplicity | RW/  RO/  WO | Description | *<timeSeriesAnnc>* Attributes |
| --- | --- | --- | --- | --- |
| *resourceType* | 1 | RO | See clause 9.6.1.3. | NA |
| *resourceID* | 1 | RO | See clause 9.6.1.3. | NA |
| *resourceName* | 1 | WO | See clause 9.6.1.3. | NA |
| *parentID* | 1 | RO | See clause 9.6.1.3. | NA |
| *expirationTime* | 1 | RW | See clause 9.6.1.3 | MA |
| *accessControlPolicyIDs* | 0..1 (L) | RW | See clause 9.6.1.3. | MA |
| *labels* | 0..1 (L) | RW | See clause 9.6.1.3. | MA |
| *creationTime* | 1 | RO | See clause 9.6.1.3. | NA |
| *lastModifiedTime* | 1 | RO | See clause 9.6.1.3. | NA |
| *announceTo* | 0..1 (L) | RW | See clause 9.6.1.3. | NA |
| *announcedAttribute* | 0..1 (L) | RW | See clause 9.6.1.3. | NA |
| *dynamicAuthorizationConsultationIDs* | 0..1 (L) | RW | See clause 9.6.1.3. | OA |
| *creator* | 1 | RO | See clause 9.6.1.3. | NA |
| *maxNrOfInstances* | 0..1 | RW | Maximum number of direct child *<timeSeriesInstance>* resources in the <*timeSeries*> resource. | OA |
| *maxByteSize* | 0..1 | RW | Maximum size in bytes of data that is allocated for the *<timeSeriesInstance>* resource for all direct child*<timeSeriesInstance>* resources. | OA |
| *maxInstanceAge* | 0..1 | RW | Maximum age of a direct child *<timeSeriesInstance>* resource in the <*timeSeries*> *resource*. The value is expressed in seconds. | OA |
| *currentNrOfInstances* | 1 | RO | Current number of direct child *<timeSeriesInstance>* resource in the <*timeSeries*> resource. It is limited by the *maxNrOfInstances*. The *currentNrOfInstances* attribute of the <timeSeries> resource shall be updated on successful creation or deletion of direct child < *timeSeriesInstance* > resource of <timeSeries > resource. | OA |
| *currentByteSize* | 1 | RO | Current size in bytes of data stored in all direct child *<timeSeriesInstance>* resources of a <*timeSeries*> resource. It is limited by the *maxByteSize*. The *currentByteSize* attribute of the <timeSeries> resource shall be updated on successful creation or deletion of direct child < *timeSeriesInstance* > resource of <timeSeries > resource. | OA |
| *periodicInterval* | 0..1 | WO | If the Time Series Data is periodic, this attribute shall contain the expected amount of time between two instances of Time Series Data. | OA |
| *periodicIntervalDelta* | 0..1 | WO | If the Time Series Data is periodic, this attribute may contain delta value relative to *periodicInterval* for the purpose of detecting missing data.  The value of this attribute must be lesser than (*periodicInterval/2*). | OA |
| *missingDataDetect* | 0..1 | WO | Indicates whether the Receiver shall detect the missing Time Series Data if it is periodic. | NA |
| *ontologyRef* | 0..1 | RW | A reference (URI) of the ontology used to represent the information that is stored in the child *<timeSeriesInstance>* resources of the present *<timeSeriesData>* resource (see note). | OA |
| *missingDataMaxNr* | 0..1 | RW | Maximum number of entries in the *missingDataList* if the *periodicInterval* is set and the *missingDataDetect* is TRUE. | OA |
| *missingDataList* | 0..1(L) | RO | The list of the *dataGenerationTime* valuerepresenting the missing Time Series Data in descending order by time if the *periodicInterva*l is set and the *missingDataDetect* is TRUE. | OA |
| *missingDataCurrentNr* | 0..1 | RO | Current number of the missing Time Series Data in the *missingDataList*. | OA |
| *missingDataDetectTimer* | 0..1 | RW | The *missingDataDetectTimer* is a duration after which a <timeSeriesInstance> shall be considered missing by the hosting CSE.  If *periodicIntervalDelta* is present, the value of this attribute must be greater than *periodicIntervalDelta.* | OA |
| *contentInfo* | 0..1 | WO | This attribute contains information to understand the contents of the *content* attribute of <timeSeriesInstance>. It shall be composed of two mandatory components consisting of an Internet Media Type (as defined in the IETF RFC 6838 [i.36]) and an encoding type. In addition, an optional content security component may also be included. The format of this attribute is defined in oneM2M TS‑0004 [3].  This attribute should be used to represent the content information of the *content* attribute of child <*timeSeriesInstance*> resources so that AEs can understand the content. | OA |
| NOTE: The access to this URI is out of scope of oneM2M. | | | | |

### ------------------------------------End of change 1---------------------------------------------