|  |
| --- |
|  |

|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* | SDS 40 |
| Source:\* | Bob Flynn, Convida Wireless , Bob.Flynn@convidawireless.com |
| Date:\* | 2019-05-20 |
| Reason for Change/s:\* | Optimization to time-series reporting procedure |
| CR against: Release\* | Rel-4 |
| 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 v4.1.0 |
| Clauses \* | 10.2.4.29 |
| 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

* Optimization to time series missing data reporting procedure

In the “Procedure for Time Series Data Detecting and Reporting”, specification mentions “If the timer expires, the timer is restarted and the missing data points counter is reset back to 0”. The statement should be modified to “If the timer expires, the missing data points counter is reset back to 0 and the timer is restarted **upon detection of next missing data**”.

The reason for the modification is to optimize the procedure by not running the timer continuously following the first missing data. The optimization comes handy when there is no more missing data detected post timer expiry which otherwise results in unnecessarily running the timer.

* Notification upon subscription delete under *<timeSeries>*

In the “Procedure for Time Series Data Detecting and Reporting”, specification mentions “a final NOTIFY request is sent out with the current number of missing data points and the timer is stopped”. The statement should be modified to “a final NOTIFY request is sent out with the current number of missing data points, **missing data list** and the timer is stopped”.

The reason for the modification is to keep the NOTIFY procedure uniform. For NOTIFY procedure in scenarios other than when the subscription under <timeSeries> is deleted, specification mentions that NOTIFY request will be sent with “current number of missing data” and “missing data list”. However, in scenario of subscription delete under <timeSeries>, specification mentions only “current number of missing data” to be included.

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

#### 10.2.4.29 Procedure for Time Series Data Detecting and Reporting

In the case that the *periodicInterval* is set and the *missingDataDetect* is TRUE, the Hosting CSE shall monitor the Time Series Data based on its *periodicInterval*. When the Hosting CSE detects a missing data point, the *dataGenerationTime* of the missing data point is inserted into the *missingDataList* attribute and the *missingDataCurrentNr* shall be increased by one. When the *missingDataCurrentNr* reaches the *missingDataMaxNr,* the oldest *dataGenerationTime* shall be removed from *missingDataList* to enable the insertion of the new missing data point information.

When an AE wants to be informed of the number of missing data points in a given renewable time duration, the AE should request the creation of a *<subscription>* resource and set the *missingData* in the *eventNotificationCriteria* conditions to specify the reporting policy. This enables the AE to keep track of the number of missing data points and the corresponding time-stamps over a predefined but renewable duration (i.e. the “window duration” of the *missingData condition*).

When the Hosting CSE reports missing data points, it shall check the *missingData* condition in the subscription resources created for that purpose.

When the first missing data point is detected (i.e. a detection of the first discontinuous time-stamp) following the creation of a subscription, the Hosting CSE shall start a timer associated with that subscription and start counting the number of missing data points. The timer is set according to the “window duration” in the subscription’s *missingData*  condition. The reporting policy is governed by the rules below:

* If the total number of missing data points becomes equal to the “minimum specified missing number of the Time Series Data” specified in the subscription’s *missingData* conditionbefore the timer expires, a NOTIFY request shall be sent including the "number of missing data points" that have been detected since the start of the subscription’s timer. The missing data points counter shall continue counting while the timer continues to run (since it did not expire). A similar NOTIFY request shall be sent for each subsequent missing data point detected until the timer expires (see next bullet for behavior when the timer expires).
* If the timer expires, the missing data points counter is reset back to 0. The timer is restarted upon detection of next missing data. The reset of the timer and the missing data points counter upon timer expiry shall continue until such time as the subscription is cancelled or terminated.
* If no missing data points have been detected at all during the life time of a subscription, then no timer shall be started at all. But once a timer is started triggered by the first missing data point, then the above rules in the previous bullets shall apply.

Figure 10.2.4.29-1depicts the above rules.



Figure 10.2.4.29-1: Time Series Data Detecting and Reporting Mechanism

T1: when the first missing data point is detected, the timer is started and the missing data points counter is set to a value of 1.

T2: the NOTIFY Request is sent when the value of the missing data points counter becomes equal to the value in the *missingData* attribute*.*

T3: the NOTIFY Request is sent again because the missing data points counter is greater than the value in the *missingData* attribute.

T4: at the end of the “window duration” the missing data points counter is reset back to 0.

T5: the “window duration” timer is restarted since amissing data point is detected and the missing data points counter is set to a value of 1.

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