|  |
| --- |
|  |

|  |
| --- |
| CHANGE REQUEST |
| Meeting ID:\* |  SDS #50 |
| Source:\* | Andreas Kraft, DT, A.Kraft@telekom.de Andreas Neubacher, DT, Andreas.Neubacher@magenta.at Bob Flynn, Exacta GSS, Bob.Flynn@exactagss.com |
|  | 2021-05-20 |
| Reason for Change/s:\* | Clarification for <timeSeries> CREATE procedure (TS-0001) (R2) |
| CR against: Release\* | Release 2 |
| CR against: WI\* | [ ]  Active WI-xxxx[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-0001, 2.28.0 |
| Clauses \* | 10.2.39 |
| Type of change: \* | [x]  Editorial change[ ]  Bug Fix or Correction[ ]  Change to existing feature or functionality[ ]  New feature or functionalityOnly ONE of the above shall be ticked |
| Impacted other TS/TR(s) |  |
| 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 [x]  |
| Template Version: January 2017 (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 CR is a mirror CR for SDS-2021-0137 and proposes an editorial correction for TS-0001 R2.

SDS-2021-0137 introduces a clarification for adding missing <timeSeriesInstance> resources to the <timeSeries>’s *missingDataList* and updating of the *missingDataCurrentNr* attribute. The same paragraph also exists in TS-0001, clause 10.2.39, and must be updated here as well.

In addition, a few typos are corrected in the same clause (marked yellow).

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of Change 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 10.2.39 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* exceeds the *missingDataMaxNr,* the oldest *dataGenerationTime* shall be removed from *missingDataList* to enable the insertion of the new missing data point information and *missingDataCurrentNr* shall be set to the value of *missingDataMaxNr*.

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 timestamps 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 susbcription’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 behaviour when the timer expires).
* If the timer expires, the timer is restarted and its missing data points counter is reset back to 0.
* The renewal of the timer and the missing data points counter upon timer expiry shall continue until such time as the subscription is cancelled or terminated. Once a subscription is terminated, a final NOTIFY request is sent out with the current number of missing data points and the timer is stopped.
* If no missing data points have been detected at all during the lifetime 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 10.2.39-1depicts the above rules.



Figure 10.2.39-1: Time Series Data Detecting and Reporting Mechanism

T1: the timer is started and the subscription begins counting the number of missing data points.

T2: a NOTIFY Request is sent because the total number of missing data points becomes equal to the "minimum specified missing number of the Time Series Data" in the subscription’s *missingData* condition*.*

T3: a NOTIFY Request is sent as there is another missing data point and the total number has exceeded the “minimum specified missing number of the Time Series Data”.

T4: the timer is restarted and the missing data points counter is reset back to 0.

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of Change 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*