|  |
| --- |
|  |

|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* | RDM #46 |
| Source:\* | Cyrille Bareau, Orange, [cyrille.bareau@orange.com](mailto:cyrille.bareau@orange.com)  Marianne Mohali, Orange, [marianne.mohali@orange.com](mailto:marianne.mohali@orange.com)  Andreas Kraft, DT, [Andreas.Kraft@t-systems.com](mailto:Andreas.Kraft@t-systems.com) |
| Date:\* | 2020-07-21 |
| Reason for Change/s:\* | See the introduction below |
| CR against: Release\* | Release 4 |
| CR against: WI\* | Active WI-0099  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 v. 4.4.1 |
| Clauses \* | Modified clauses: 5.8.8 |
| 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 |
| Impacted other TS/TR(s) |  |
| 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 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 draft proposes a new approach for logging events from a device, based on two recently introduced features:

1. Device Management with SDT, using flexContainers.
2. History management, using flexContainerInstances.

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of change 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 5.8.8 dmEventLog

This ModuleClass provides DM capabilities to control and monitor event logs of the device.

Table 5.8.8-1 Actions of dmEventLog ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Arguments | Optional | Description |
| none | retrieveLog | start: xs:datetime  end: xs:datetime | true | Upload from the device the logging data between ‘start’ and ‘end’.  ‘start’ must be a date before ‘end’, and is optional. The default is beginning of time.  ‘end’ must be a date after ‘start’ and is optional. The default is the timestamp of the last available log entry. |

This action, if provided, requests the IPE to read logging data on the device. This log is then stored in the ‘data’ datapoint. It is only valid when the ‘enabled’ datapoint is *true*. The *start* and *end* arguments are only indications of the timeframe for the log retrieval. If a target device can deliver only partial logs for a given timeframe, for example when the *start* argument is too far in the past and logs are not available for that time anymore, then the device shall deliver logs from the earliest available point in time on.

Table 5.8.8-2 DataPoints of dmEventLog ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Description |
| type | m2m:logTypeId | R | false |  | The type of the log (e.g. security log, system log…). |
| data | xs:string | R | false |  | Raw data of *last* event. No format specified. |
| status | m2m:logStatus | R | false |  | The current status of the logging process (Started, Stopped, Error, etc.) |
| enabled | xs:boolean | RW | false |  | Start / stop logging. |

For devices using the dmEventLog ModuleClass, the following rules apply:

* The actual logging process on the device (if any), and the retrieval of device logging data by the IPE, are out of scope of this document.
* Instances of this module should only be created by the IPE (one per log type supported by the device for instance).
* The IPE can create a [dmEventLog] instance with *status* datapoint ‘NotPresent’ for a given log type, to indicate that this log type is not supported by the device. Otherwise *status* should have value ‘Started’ (resp. ‘Stopped’) if the *enabled* datapoint is set to *true* (resp. *false*). The *status* datapoint can be given ‘Error’ value if the log processing dysfunctions.
* The IPE should use the <*flexContainerInstance*> history mechanism (see TS-0001 § 9.6.59) by setting on [dmEventLog] at least one attribute *maxNrOfInstances*, *maxByteSize* or *maxInstanceAge*. Then for each log event read by the IPE from the device, and if the *enabled* datapoint has value *true*, a <*flexContainerInstance*> resource shall be created, child of this module <*flexContainer*>. The [dmEventLog] module itself just contains the *last* logged event from the device for this log type.
* The [dmEventLog] <*flexContainer*>, and therefore its <*flexContainerInstance*> children resources, should have a *dataGenerationTime* custom attribute that indicates the time the event was logged *on the device* (see Rule 2-5 in section 6.2.3).
* When the *enabled* datapoint is set to *false*, the IPE shall set the *status* datapoint to ‘Stopped’ and shall not modify the *data* datapoint of the module, and therefore shall not create any <*flexContainerInstance*> child resource.

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