|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* | SDS 42 |
| Source:\* | Neeta Meshram ([neeta@cdot.in](mailto:neeta@cdot.in)), Poornima Trivedi (poornima@cdot.in), Anupama Chopra(anupama@cdot.in) C-DOT |
| Date:\* | 2019-09-23 |
| Reason for Change/s:\* | See the Introduction |
| CR against: Release | Release-3 |
| CR against: WI\* | Active  MNT maintenance  Is this a mirror CR? Yes  No  mirror CR number: (Note to Rapporteur - use latest agreed revision)  STE Small Technical Enhancements  Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0004-Service\_Layer\_Core\_Protocol-V3\_13 |
| Clauses \* | 7.4.8.2.1, 7.4.8.2.3 |
| 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 |  |
| 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.

If this is a correction, and the change applies to previous releases, a separate “mirror CR” should be posted at the same time as 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. Include any changes to references, definitions, and abbreviations in the same deliverable.

Follow the drafting rules.

All pictures must be editable.

Check spelling and grammar.

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 proposed new clause is 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 the 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

*EventNotificationCriteria* attribute of <subscription> resource indicates the event criteria for which notification is to be generated. One such event criteria is *missingData* as provided in TS-0001 by following table:

Table-9.6.8-3: *eventNotificationCriteria* conditions

| Condition tag | Multiplicity | Matching condition |
| --- | --- | --- |
| *createdBefore* | 0..1 | The *creationTime* attribute of the resource is chronologically before the specified value. |
| *createdAfter* | 0..1 | The *creationTime* attribute of the resource is chronologically after the specified value. |
| *modifiedSince* | 0..1 | The *lastModifiedTime* attribute of the resource is chronologically after the specified value. |
| *unmodifiedSince* | 0..1 | The *lastModifiedTime* attribute of the resource is chronologically before the specified value. |
| *stateTagSmaller* | 0..1 | The *stateTag* attribute of the resource is smaller than the specified value. |
| *stateTagBigger* | 0..1 | The *stateTag* attribute of the resource is bigger than the specified value. |
| *expireBefore* | 0..1 | The *expirationTime* attribute of the resource is chronologically before the specified value. |
| *expireAfter* | 0..1 | The *expirationTime* attribute of the resource is chronologically after the specified value. |
| *sizeAbove* | 0..1 | The *contentSize* attribute of the *<contentInstance>* resource is equal to or greater than the specified value. |
| *sizeBelow* | 0..1 | The *contentSize* attribute of the *<contentInstance>* resource is smaller than the specified value. |
| *notificationEventType* | 0..6 | The type of event that shall trigger a notification. If multiple *notificationEventType* tags are present, a notification shall be triggered if any of the configured events occur. Note that not all permutations of event type are meaningful. Possible notification event type values are:   1. Update to attributes of the subscribed-to resource 2. Deletion of the subscribed-to resource, 3. Creation of a direct child of the subscribed-to resource, 4. Deletion of a direct child of the subscribed-to resource 5. An attempt to retrieve a <*contentInstance*> direct-child-resource of a subscribed-to <*container*> resource is performed while this <*contentInstance*> child resource is an obsolete resource or the reference used for retrieving this resource is not assigned. This retrieval is performed by a RETRIEVE request targeting the subscribed-to resource with the Result Content parameter set to either "child-resources" or "attributes+child-resources". This value for the *eventNotificationType* tag implies that the subscribed-to resource shall be an <*container*> resource. Otherwise this setting is not valid. 6. Trigger Received targeting the MN/ASN-AE associated with the <AE> parent resource. This implies that the subscribed-to resource shall be an <*AE*> resource instance. Otherwise this setting is not valid. 7. Update to attributes of thesubscribed-to resource with blocking of the triggering UPDATE operation. For this *eventNotificationType* value setting, only one single Notification Target shall be present in the *notificationURI* attribute – see *notificationURI* attribute definition. This value for the *eventNotificationType* tag shall not be combined with any other *eventNotificationType* tag value. This value for *notificationEventType* establishes a subscription that is triggered for the same events as for the value “Update to attributes of the subscribed-to resource”. However, upon occurrence of a triggering UPDATE operation that has been validated and results in an authorized UPDATE operation, the triggering UPDATE operation shall be blocked by the Hosting CSE until a notification request was sent out and a corresponding response message was received or a timeout happens. When the response status code of the notification response message indicates a successful notification reception in combination with a successful notification action taken by the Notification Target entity, the triggering UPDATE operation shall be completed with a successful update of the targeted attribute(s). If the notification response message indicates an unsuccessful notification reception or a successful notification reception with unsuccessful notification action by the targeted entity or times out, the blocked UPDATE operation shall be completed with no success and no change of the targeted attribute(s). For any subscribed-to resource there shall exist a maximum of one subscription with this setting of *notificationEventType*. All other notification policies shall not be allowed when this setting of *notificationEventType* is used. The *notificationContentType* shall be “modified attibutes”. When an UPDATE operation has been blocked due to triggering this type of notification, any other occurring UPDATE or DELETE requests to the same resource shall be handled only after the blocked UPDATE operation has been completed.   The other conditions in *eventNotificationCriteria* conditions apply within the scope of the selected *notificationEventType.*  For example, if notificationEventType is "Creation of a direct child of the subscribed-to resource" then other *eventNotificationCriteria* conditions is applied to the direct child resources of the subscribed-to resource.  If this condition is not specified, the default value is "Update to attributes of the subscribed-to resource". This default value shall apply only if *operationMonitor* is not present in the resource.  The notion of "obsolete resource" is defined in clause 9.6.1.3.2 (Common attributes). |
| *operationMonitor* | 0..n | The operations and/or the Originators accessing the subscribed-to resource matches with the specified value. It allows monitoring which operation and/or which Originator is attempting to the access subscribed-to resource regardless of whether the operation is performed. This feature is useful to detect AEs that send requests to a subscribed-to resource and that result in a successful or failure response. Possible arguments are operation(s) (e.g.: CREATE, RETRIEVE, UPDATE, DELETE, NOTIFY) and/or Originator identifier(s).  If a set of Originator identifier(s) is included in this tag and no operations are listed, any operations initiated from any of the indicated Originator(s) shall trigger a notification.  If a set of operation(s) is included in this tag and no Originator identifier, any of the listed operations shall trigger a notification.  If both, a set of Originator identifiers and a set of operations are listed, then any of the listed operations initiated from any of the listed Originators shall trigger the notification. |
| *attribute* | 0..n | A list of attribute names of a subscribed-to-resource. This list is only applicable when *notificationEventType* has a value of "Update to attributes of the subscribed-to resource". or “Update to attributes of the subscribed-to resource with blocking of the triggering UPDATE operation”.  If this list is present, then it is used to specify a subset of a subscribed-to resource's attributes for which updates shall result in a notification. If ANY attribute specified on this list is updated, then a notification shall be generated. If an attribute that is not specified in this list is updated, then a notification shall not be generated.  If this list is not presented, then the default attribute list is the full set of a subscribed-to resource's attributes. If ANY attribute of a subscribed-to resource is updated, then a notification shall be generated. |
| *childResourceType* | 0.. 1 (L) | A list of resource types. This list is only applicable when *notificationEventType* has a value of "Creation of a direct child of the subscribed-to resource ".  If this list is present, then it is used to specify a subset of resource type for direct child resource of which creation shall result in a notification. If ANY resource type specified on this list is created, then a notification shall be generated. If a resource type that is not specified in this list is created, then a notification shall not be generated.  If this list is not present, then the default resource type list is the full set of a direct child resource. |
| *missingData* | 0..1 | The *missingData* includes two values: a minimum specified missing number of the Time Series Data within the specified window duration, and the window duration. The condition only applies to subscribed-to resources of type *<timeSeries>*. If this attribute is modified by an UPDATE the associated timer/counter are stopped and restarted with the new values.  The first detected missing data point starts the timer associated with the window duration.  The window duration is restarted upon its expiry until such time as the entire subscription is terminated or not refreshed. More details about NOTIFICATIONS related to data reporting is found in section 10.2.4.29 |
| *filterOperation* | 0..1 | Indicates the logical operation (AND/OR) to be used for the condition tags *createdBefore, createdAfter, modifiedSince, unmodifiedSince, stateTagSmaller, stateTagBigger, expireBefore, expireAfter, sizeAbove, sizeBelow*. The default value is logical AND. |

As per highlighted portion above, *missingDataCondition* only applies if subscribed-to-resource is of type <timeSeries>.

This CR proposes to add validation when *missingData* is provided by originator, subscribed-to-resource shall be <timeSeries> resource.

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

#### 7.4.8.2 <subscription> resource specific procedures for CRUD operations

##### 7.4.8.2.1 Create

***Originator:***

No change from the generic procedures in clause Error: Reference source not found.

***Receiver:***

The following are additional Hosting CSE procedures to the generic resource handling procedures (Error: Reference source not found in clause Error: Reference source not found). The additional procedures shall be inserted from Recv-6.2 to Recv-6.5 as below.

Recv-6.2. The following step is in addition to the procedures defined in clause 7.3.3.1:

Check if the subscribed-to resource, addressed in ***To*** parameter in the Request, is subscribable. Subscribable resource types are defined in TS-0001 [Error: Reference source not found]; they have <subscription> resource types as their child resources. If it is not subscribable, the Hosting CSE shall return the Notify response primitive with a ***Response Status Code*** indicating "TARGET\_NOT\_SUBSCRIBABLE" error.

Recv-6.3 The following step is in addition to the procedures defined in clause 7.3.3.15:

Check if the Originator has privileges for retrieving the subscribed-to resource. If the Originator does not have the privilege, the Hosting CSE shall return the response primitive with ***Response Status Code*** indicating "ORIGINATOR\_HAS\_NO\_PRIVILEGE" error.

Recv-6.4 The following steps are in addition to the procedures defined in clause 7.3.3.3:

1. Check if the *notificationEventType* is set to "Blocking\_Update".
   1. If the subscribed-to resource already has a subscription with this *notificationEventType* the Hosting CSE shall return the response primitive with ***Response Status Code*** indicating "BLOCKING\_SUBSCRIPTION\_ALREADY\_EXISTS" error if more than one notification of this type could be sent.
   2. If there is more than one *notificationURI* specified, the Hosting CSE shall return the response primitive with ***Response Status Code*** indicating "BAD\_REQUEST" error.
   3. If any other attributes of the <subscription> resource are specified the Hosting CSE shall return the primitive with ***Response Status Code*** indicating "BAD\_REQUEST" error.
   4. If any condition tag of the *eventNotificationCriteria* attribute other than *attribute* condition tag is specified, the Hosting CSE shall return the response primitive with ***Response Status Code*** indicating "BAD\_REQUEST" error.
2. If any of the *notificationURI* entries are not the Originator, the Hosting CSE may send a Subscription Verification request primitive to each of them as described in clause 7.5.1.2.3.
   1. If the Hosting CSE cannot send one or more Subscription Verification request primitives, the Hosting CSE shall return the Create <subscription> response primitive with a ***Response Status Code*** indicating "SUBSCRIPTION\_VERIFICATION\_INITIATION\_FAILED" error.
   2. If the Hosting CSE sent all the Subscription Verification request primitives, the Hosting CSE shall check if each Notify response primitive contains a ***Response Status Code*** indicating "OK". If not, the Hosting CSE shall return the Create <subscription> response primitive containing the ***Response Status Code*** indicating "SUBSCRIPTION\_VERIFICATION\_INITIATION\_FAILED" error.
3. If the *associatedCrossResourceSub* is provided, check that the Hosting CSE ID value in the *associatedCrossResourceSub* is the same as the ***From*** parameter of the request. If not, return the response primitive with a ***Response Status Code*** indicating "BAD\_REQUEST".

Recv-6.5: The following steps are in addition to the procedures defined in clause 7.3.3.5:

1. If the Originator provides a value of *childResourceType* which is not a valid child of the subscribed-to resource, the request shall be rejected with a “BAD\_REQUEST” ***Response Status Code.***
2. If the Originator provides *missingData,*check that subscribed-to-resource is of type <timeSeries>. If not, request shall be rejected with "BAD\_REQUEST" *Response Status Code.*
3. If both the *notificationEventType* and *operationMonitor* are present in the Request, the request shall be rejected with a "BAD\_REQUEST" ***Response Status Code.***
4. If the Originator does not provide *notificationContentType*, the Hosting CSE shall set it as "all attributes".
5. If the *notificationURI* is not the Originator, the Hosting CSE shall store the Originator ID as the <subscription> resource's *creator* attribute.
6. If the *batchNotify* attribute is present in the Request but *batchNotify*/*duration* is not provided by the Originator, the Hosting CSE shall set the value of *batchNotify*/*duration* to the default duration as given by the M2M Service Provider.

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

### -----------------------Start of change 2-------------------------------------------

##### 7.4.8.2.3 Update

***Originator:***

The following change from the generic procedures in clause Error: Reference source not found.

Orig-1.0: The originator shall not specify *notificationEventType* set to "Blocking\_Update".

***Receiver:***

The following are additional Hosting CSE procedures to the generic resource handling procedures in clause Error: Reference source not found.

Recv-6.4: The following steps are in addition to the procedures defined in clause 7.3.3.4:

1. Check if the notificationEventType is set to "Blocking\_Update". If so, the Hosting CSE shall return the response primitive with ***Response Status Code*** indicating "BAD\_REQUEST" error.

Recv-6.5. The following steps are in addition to the procedures defined in clause 7.3.3.7:

1. If the Originator provides a value of *childResourceType* which is not a valid child of thesubscribed-to resource,the request shall be rejected with a “BAD\_REQUEST” ***Response Status Code***.
2. If the Originator provides *missingData,*check that subscribed-to-resource is of type <timeSeries>. If not, request shall be rejected with "BAD\_REQUEST" *Response Status Code.*
3. If the UPDATE operation would result in both *operationMonitor* and *notificationEventType* being present in the resource, the request shall be rejected with a “BAD\_REQUEST” ***Response Status Code.***
4. Check if a new *associatedCrossResourceSub* is provided. If so, check that the Hosting CSE ID value in the *associatedCrossResourceSub* is the same as the ***From*** parameter of the request.
5. If a <crossResourceSubscription> Hosting CSE ID is removed from *associatedCrossResourceSub*, the Hosting CSE shall send a Notify request for Subscription Deletion using the procedures in clause 7.5.1.2.4 to the <crossResourceSubscription> Hosting CSE*.*

### -----------------------End of change 2---------------------------------------------

CHECK LIST

* Does this Change Request include an informative introduction containing the problem(s) being solved, and a summary list of proposals.?
* Does this CR contain changes related to only one particular issue/problem?
* Have any mirror CRs been posted?
* Does this Change Request make **all** the changes necessary to address the issue or problem? E.g. A change impacting 5 tables should not include a proposal to change only 3 tables?Does this Change Request follow the drafting rules?
* Are all pictures editable?
* Have you checked the spelling and grammar?
* Have you used change bars for all modifications?
* Does the change 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.)
* Are multiple changes in this CR 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.?