|  |
| --- |
|  |

|  |
| --- |
| CHANGE REQUEST |
| Meeting ID:\* | SDS 55 |
| Source:\* | Bob Flynn, Exacta, bob.flynn@exactagss.com Miguel Angel Reina Ortega, ETSI, MiguelAngel.ReinaOrtega@etsi.org Andreas Kraft, DT, A.Kraft@telekom.de |
| Date:\* | 2022-07-15 |
| Reason for Change/s:\* | Fixes for crossResourceSubscription |
| CR against: Release\* | Rel-4 |
| CR against: WI\* | [x]  Active < WI-0077> [ ]  MNT maintenance / < Work Item number(optional)>Is this a mirror CR? Yes [x]  No [ ] mirror CR number: SDS-2021-0088R04-TS-0004\_Fixes\_for\_crossResourceSubscription\_R3[ ]  STE Small Technical Enhancements / < Work Item number (optional)>Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0004 v4.10.0 |
| Clauses \* | 7.4.58.1, 7.4.58.2.1, 7.4.58.2.3, 7.4.58.2.4, 7.5.1.2.18, 7.5.1.2.21, 7.4.8.2.3, 8.2.3 |
| 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 |
| Other TS/TR(s) impacted | None |
| 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 [ ]  |
| 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

During implementation of test cases for crossResourceSubscription functionality, some misalignments were found between TS-0001 and TS-0004. This CR proposes some fixes for crossResourceSubscription functionality in order to align TS-0001 and TS-0004.

In addition, some further missing checks during the crossResourceSubscription create and notification procedures. Fixes for <subscription> update procedures. New clause for notification of cross resource subscription deletion and the following fixes:

- Correcting CRS procedures

- Clarifying subscription procedures

- Clarifying deletion notification procedures for subscription and CRS

- Adding a new attribute to crossResourceSubscription to store created subscriptions

- Adding expirationTime attribute to subscriptions during the creation



<http://tools.etsi.org/plant/png/rLPTRzem57tFhxYZ3stLPXlHzXA5I0nIQTBMJj0-JACFJkA0jM17Zkd7Vlrinq0SM9PjK6ay889FplquvzxaT45I31bZ8cPFpYknm4iCHOu8TPm2SK4IKY0ge3UCUe1A64OG1C0ucHEA1ECGXZOkI5ZEkF-u80A7esaqezO_06z8wX8AeEaa04xYTzvRxxrDCvfE362Y0d9SiXLFy4wuwTq7w6ka_DMNq5QajqPw3QHpUj2FCnZ0Y6CaC0Io0bdrOE0uidTkg4hsOJI9XhTHTJYZ6VTHd3YKINoECm4dZ2Cwnxug_pNdIC00la6vYlkNwjkxTnpLETSDRJu9dz62imIdAuxt4LRf67WfP8DAYsi25nThEfqOgCmK80XSLm5gSDto1jw2Au2Dztw7ztIV0jUgIqfLC5fYvnX2tNqPgnWvOuKRe-Gxo0R6ELusDJCutUFkYddtCth6yP8zO2tWULEwEcq7wGnyHphLQX_6qQUesLZNd5tVt7wy-he-KqPAVS5NUsDLFC2oIkCSexG4mUJLUCCZ1pfPWDmOT7kk35JTsopJC7Ojqy3tbUcrbEcrbAavAWRRSPlaE1OhJfqKbuApP-F6-d3iZxmUZFLzso49hTPlRMmDI4lulZKCHlgZ3eD9fHBXZEB7Spq6A17aGQqpLTlcGcTOl_AiWQbpRJfsD5lyeVi5yoLZ-nIGEMLSZMqAyfHa9462oDRNnMqiErkyVx3jWV9hMFqmtgvvScFT5_2ap3c4_j-uSjUK2gGhVNh-iQ-rASaopB5ySRYQt7o6deJrVk7TYghTl3hrpkPLENNStI-uPJMBV8nn5UfIh_gc4-ik1FDsPMJhulO3c_r_sUkD6Ve9>



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

##### 7.4.58.2.3 Update

***Originator:***

No change from the generic procedures in clause 7.2.2.1.

***Receiver:***

The following are changes to the receiver procedures described in clause 7.2.2.2.

1. Recv-6.5: The following steps are in addition to the generic Update procedures defined in clause 7.3.3.7:
2. If any of *timeWindowSize* or *timeWindowType* attributes is updated, the receiver shall restart the time window as described in clause 7.4.58.2.1.
3. If the *notificationStatsEnable* attribute in the resource is true and the *notificationStatsEnable* attribute in the request is false, the Hosting CSE shall stop collecting notification statistics for the *<*crossResource*S*ubscription*>* resource. The Hosting CSE shall maintain the current value of the *notificationStatsInfo* attribute.
4. If the *notificationStatsEnable* attribute in the resource is false and the *notificationStatsEnable* attribute in the request is true, the Hosting CSE shall update the value of the *notificationStatsEnable* attribute in the resource to true, delete any values stored in the *notificationStatsInfo* attribute of the resource and then start recording notification statistics.

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

## ----------------------- Start of Change 2--------------------------------------------

##### 7.4.58.2.4 Delete

***Originator:***

No change from the generic procedures in clause 7.2.2.1.

***Receiver:***

The following are changes to the receiver procedures described in clause 7.2.2.2:

1. Recv-6.5: The following steps are in addition to the generic Delete procedures defined in clause 7.3.3.8:
2. The Hosting CSE shall stop the time window timer if running.
3. If this procedure was caused by a subscription deletion notification (see clause 7.5.1.2.4), then the following steps shall exclude the subscription reference indicated in the subscription deletion notification.
4. The Hosting CSE shall delete the previously created <subscription> child resource of each target resource indicated in the *regularResourcesAsTarget* attribute. Target subscriptions are indicated in *regularResourcesAsTargetSubscriptions* attribute of the <crossResourceSubscription>. If this procedure is triggered by a DELETE operation, then the Receiver shall use the ***From***of the current request for these requests. Otherwise, the Receiver shall use the creator of this <crossResourceSubscription>.
5. The Hosting CSE shall retrieve and update each <subscription> resource indicated in the subscriptionResourcesAsTarget attribute using the procedure in clause 7.4.8.2.3 to remove the resource identifier of this <crossResourceSubscription> from the <subscription> resource's associatedCrossResourceSub and notificationURI attributes. If this procedure is triggered by a DELETE operation, then the Receiver shall use the From of the current request for these requests. Otherwise, the Receiver shall use the creator of this <crossResourceSubscription>.
6. Any failure in steps c) or d) shall be ignored.

f) The Hosting CSE shall send a Notify request for Cross Resource Subscription Deletion using the procedures in clause 7.5.1.2.21.

-------------------------------------------------- End of Change 2---------------------------------------

## ----------------------- Start of Change 3--------------------------------------------

##### 7.4.58.2.1 Create

***Originator:***

The following are changes to the Originator procedures described in clause 7.2.2.1:

1. Orig-1.0 When composing a request primitive, the Originator shall include *regularResourcesAsTarget* and/or *subscriptionResourcesAsTarget* attributes in the resource representation of the <crossResourceSubscription> in the content of the primitive. If *regularResourcesAsTarget* attribute is included, *eventNotificationCriteriaSet* attribute shall be included. If *eventNotificationCriteriaSet* contains only one *eventNotificationCriteria*, this *eventNotificationCriteria* shall be applied to all regular resources included in *regularResourcesAsTarget* attribute; otherwise, *eventNotificationCriteriaSet* shall contain the same number of *eventNotificationCriteria* elements as the number of regular target resources contained in *regularResourcesAsTarget* and each eventNotificationCriteria element shall be sequentially applied to corresponding target resource as listed in the *regularResourcesAsTarget*.

***Receiver:***

The following are changes to the receiver procedures described in clause 7.2.2.2:

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

a) The request shall be rejected with a "BAD\_REQUEST" ***Response Status Code*** if at least one of *regularResourcesAsTarget* or *subscriptionResourcesAsTarget* attributes is not present in the request.

b) If *subscriptionResourcesAsTarget* is included, the Hosting CSE shall retrieve and update each <subscription> resource indicated in *subscriptionResourcesAsTarget* by issuing an UPDATE request to the <subscription> resource host as follows:

i) In the UPDATE request, the receiver shall use the ***From*** *parameter* from the current CREATE request.

ii) *associatedCrossResourceSub* attribute shall be updated by adding the resource identifier of this <crossResourceSubscription> resource being created.

iii) *notificationURI* attribute shall be updated by adding the resource identifier of this <crossResourceSubscription> resource being created. The Hosting CSE shall properly reply to a potential subscription verification request.

iv) If any <subscription> for a target resource cannot be successfully updated or the <crossResourceSubscription> CREATE Request timeout is exceeded, the receiver shall send an unsuccessful response with a "CROSS\_RESOURCE\_OPERATION\_FAILURE" ***Response Status Code*** to the Originator; the Hosting CSE shall also remove itself from any already successfully associated <subscription> resources using the procedures in clause 7.4.8.2.4 and also delete any already-created <subscription> resources at other target resources.

c) If *regularResourcesAsTarget* is included, the Hosting CSE shall send a CREATE <subscription> request message to each target resource indicated by *regularResourcesAsTarget*.

i) The request shall be rejected with “BAD\_REQUEST” ***Response Status Code*** if *eventNoficationCriteriaSet* attribute is not present in the request. If present, the *eventNotificationCriteriaSet* attribute shall contain either one *eventNotificationCriteria* or the same number as the number of regular target resources contained in *regularResourcesAsTarget* attribute. Otherwise, the request shall be rejected with “BAD\_REQUEST”.

ii) In the new CREATE <subscription> request, the receiver shall use the ***From***of the current CREATE request. For this <subscription> to be created:

1) *eventNotificationCriteria* attribute shall use the corresponding entry included in *eventNotificationCriteriaSet* attribute of the <crossResourceSubscription> resource representation. If *eventNotificationCriteriaSet* attribute contains only one entry, *eventNotificationCriteria* attribute shall use that entry.

2) *notificationURI* attribute shall be set to the resource identifier of this <crossResourceSubscription> resource being created. The Hosting CSE shall properly reply to a potential subscription verification request.

3) *associatedCrossResourceSub* attribute shall be set to the resource identifier of this <crossResourceSubscription> resource being created.

4) *notificationEventCat* attribute shall be set to the same value in the <crossResourceSubscription> resource representation if present. Otherwise, it shall not be set..

5) *expirationTime* attribute shall be set to the same value as the *expirationTime* attribute in the <crossResourceSubscription> resource, if present, or the value provided by the Hosting CSE for this <crossResourceSubscription> resource

iii) If any <subscription> for a target resource cannot be successfully created or the <crossResourceSubscription> CREATE Request timeout is exceeded, the receiver shall send an unsuccessful response with a "CROSS\_RESOURCE\_OPERATION\_FAILURE" ***Response Status Code*** to the Originator; the receiver shall also delete already created <subscription> resources at other target resources that were created based on the presence of *regularResourcesAsTarget*.

iv) Upon successful creation of each <subscription> resource, the resource identifier of the recently created <subscription> shall be added to the *regularResourcesAsTargetSubscriptions* attribute at the position of the corresponding entry in *regularResourcesAsTarget.*

1. Once the <crossResourceSubscription> resource is created, the Hosting CSE shall start the time window if the *timeWindowType*=PERIODICWINDOW; if *timeWindowType*=SLIDINGWINDOW, the Hosting CSE shall start the time window after the first notification is received from a Target Resource Hosting CSE.
2. If the *notificationStatsEnable* attribute is set to true, the Hosting CSE shall start recording notification statistics in the *notificationStatsInfo* attribute once the <crossResourceSubscription> resource is created.

-------------------------------------------------- End of Change 3---------------------------------------

## ----------------------- Start of Change 4--------------------------------------------

##### 7.5.1.2.18 Cross-Resource Notification

When the <crossResourceSubscription> Hosting CSE receives a notification from the Host of a <subscription> indicated in *regularResourcesAsTarget* or *subscriptionResourcesAsTarget* the <crossResourceSubscription> Hosting CSEshall perform the following steps:

1. The Hosting CSE shall send a notification response to the <subscription> resource Hosting CSE.
2. Aggregate notifications using the time window mechanism indicated by *timeWindowType* attribute of the <crossResourceSubscription> resource to determine if a cross-resource notification shall be issued:

a) The Hosting CSE shall store the received notification until the current time window expires. When the current time window expires, the Hosting CSE shall discard stored notifications:

i) If *timeWindowType* is PERIODICWINDOW then a new time window shall be started when the current time window expires.

ii) If *timeWindowType* is SLIDINGWINDOW then a new time window shall be started when the next notification is received.

1. When notifications from all target <subscription> resources occur within the required time window the Hosting CSE shall compose a cross-resource notification in a notification data object with type m2m:notification:
	1. Set *subscriptionReference* element as the URI of the <crossResourceSubscription> resource.
2. If the expirationCounter attribute is set, then it shall be decreased by one when the Originator successfully sends the Notify request primitive. If the counter equals to zero('0'), the corresponding <crossResourceSubscription> resource shall be deleted as described in 7.4.58.2.4.
3. Send the notification to the notificationURI using the procedure defined in clause 7.5.1.2.2.
4. "Wait for Response primitive" procedure.

The **Subscriber or Notification Targets** which receive cross-resource notifications from the Hosting CSE shall perform the following steps in order:

1. "Create a success response" procedure defined in clause 7.3.3.12.
2. "Send the Response primitive" procedure.

-------------------------------------------------- End of Change 4---------------------------------------

## ----------------------- Start of Change 5--------------------------------------------

#### 7.4.58.1 Introduction

The <crossResourceSubscription> resource is used to support cross-resource subscription and cross-resource notification. A cross-resource subscription is made on multiple target resources. A cross-resource notification shall be generated if and only if expected events on all target resources occur within a designated time window. When notifications from all target resources occur within a specified time window the Hosting CSE shall issue a cross-resource notification.

The detailed description can be found in clause 9.6.58 in oneM2M TS-0001 [6].

Table 7.4.58.1‑1: Data type definition of <crossResourceSubscription> resource

|  |  |  |
| --- | --- | --- |
| **Data Type ID** | **File Name** | **Note** |
| crossResourceSubscription | CDT-crossResourceSubscription.xsd |  |

Table 7.4.58.1‑2: Universal/Common Attributes of <crossResourceSubscription> resource

|  |  |
| --- | --- |
| Attribute Name | Request Optionality  |
| Create | Update |
| *@resourceName* | O | NP |
| *resourceType* | NP | NP |
| *resourceID* | NP | NP |
| *parentID* | NP | NP |
| *creationTime* | NP | NP |
| *lastModifiedTime* | NP | NP |
| *labels* | O | O |
| *accessControlPolicyIDs* | O | O |
| *expirationTime* | O | O |
| *dynamicAuthorizationConsultationIDs* | O | O |
| *custodian* | O | O |
| *creator* | O | NP |

Table 7.4.58.1‑3: Resource Specific Attributes of <crossResourceSubscription> resource

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Request Optionality  | Data Type | Default Value and Constraints |
| Create | Update |
| *regularResourcesAsTarget* | O | NP | m2m:listOfURIs | No default. At least one of those shall be present. |
| *subscriptionResourcesAsTarget* | O | NP | m2m:listOfURIs |
| *regularResourcesAsTargetSubscriptions* | NP | NP | m2m:listOfURIs | No default  |
| *timeWindowType* | M | O | m2m:timeWindowType | PERIODICWINDOW |
| *timeWindowSize* | M | O | xs:duration | No default |
| *eventNotificationCriteriaSet* | O | NP | m2m:eventNotificationCriteriaSet  | Default behaviour is notification on Update\_of\_Resource |
| *notificationEventCat* | O | NP | m2m:eventCat | No default |
| *expirationCounter* | O | O | xs:positiveInteger | No default |
| *notificationURI* | M | O | list of xs:anyURI | No default |
| *subscriberURI* | O | NP | xs:anyURI | No default |
| *notificationStatsEnable* | O | O | xs:boolean | false |
| *notificationStatsInfo* | NP | NP | m2m:setOfNotificationStatsInfo | No default |

Table 7.4.58.1‑4: Child Resources of <crossResourceSubscription> resource

|  |  |  |  |
| --- | --- | --- | --- |
| Child Resource Type | Child Resource Name | Multiplicity | Ref. to Resource Type Definition |
| <notificationTargetMgmtPolicyRef> | [variable] | 0..n | Clause 7.4.30 |
| <notificationTargetSelfReference> | ntsr | 1 | Clause 7.4.33 |
| <transaction> | [variable] | 0..n | Clause 7.4.61 |
| <schedule> | notificationSchedule | 0..1 | Clause 7.4.9 |

-------------------------------------------------- End of Change 5---------------------------------------

## ----------------------- Start of Change 6--------------------------------------------

##### 7.4.8.2.3 Update

***Originator:***

The following change from the generic procedures in clause 7.2.2.1.

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

If the Originator specifies a *missingData* condition with a duration value greater than the *periodicInterval* attribute of the <timeSeries> resource no notification on missing data points will be generated.

***Receiver:***

The following are additional Hosting CSE procedures to the generic resource handling procedures in clause 7.2.2.2.

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

1. Check if the *notificationEventType* in the request is set to "Blocking\_Update". If so, the request shall be rejected with a ***Response Status Code*** indicating a "BAD\_REQUEST" error.
2. Check if the notificationEventType is set to “Report on missing data points”. If the *missingData* attribute is not set in the target resource or provided in the request, the request shall be rejected with a ***Response Status Code*** indicating a "BAD\_REQUEST" error***.***
3. Check if the *missingData* element of eventNotificationCriteria is provided.
* If the subscribed-to resource (i.e. the resource given by the ***To*** parameter in the Request) is not a <timeSeries>, the request shall be rejected with a ***Response Status Code*** indicating a "BAD\_REQUEST" error.
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 ***Response Status Code*** indicating a “BAD\_REQUEST” error.
2. If the Originator provides *missingData*, check that the subscribed-to resource is of type <timeSeries>. If not, the request shall be rejected with a ***Response Status Code*** indicating a "BAD\_REQUEST" error.
3. If the UPDATE operation would result in both *operationMonitor* and *notificationEventType* being present in the resource, the request shall be rejected with a ***Response Status Code*** indicating a “BAD\_REQUEST” error.
4. If the *notificationContentType* is invalid for a given operation (refer to oneM2M TS-0001 [6] Table 9.6.8-4: Default and allowed values of *notificationContentType*) the request shall be rejected with a ***Response Status Code*** indicating a “BAD\_REQUEST” error.

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

1. If a <crossResourceSubscription> resource identifier 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*.*
2. If the *notificationStatsEnable* attribute in the resource is true and the *notificationStatsEnable* attribute in the request is false, the Hosting CSE shall stop collecting notification statistics for the *<*subscription*>* resource. The Hosting CSE shall maintain the current value of the *notificationStatsInfo* attribute.
3. If the *notificationStatsEnable* attribute in the resource is false and the *notificationStatsEnable* attribute in the request is true, the Hosting CSE shall update the value of the *notificationStatsEnable* attribute in the resource to true, delete any values stored in the *notificationStatsInfo* attribute of the resource and then start recording notification statistics.
4. Check if the *pendingNotification* attribute is being removed by the request or is being changed from "sendAllPending" to "sendLatest". If the *pendingNotification* attribute is being removed, then all cached pending Notify request primitives for the subscription resource shall be removed. If the *pendingNotification* attribute is being changed from "sendAllPending" to "sendLatest", then all cached pending Notify request primitives except the latest notification for the subscription resource shall be removed.

-------------------------------------------------- End of Change 6---------------------------------------

## ----------------------- Start of Change 7--------------------------------------------

##### 7.5.1.2.21 Notification of Cross Resource Subscription Deletion

***Originator:***

When the <crossResourceSubscription> resource is deleted the Originator shall send Notify request primitives to the <crossResourceSubscription> resource's subscriberURI if it is configured:

a) subscriptionDeletion element of the notification data object set as true.

b) subscriptionReference element of the notification data object set as the resource identifier of the <crossResourceSubscription> resource.

c) The ***To*** parameter shall be set to the entity indicated in *subscriberURI*.

-------------------------------------------------- End of Change 7---------------------------------------

## ----------------------- Start of Change 8--------------------------------------------

##### 7.4.8.2.1 Create

***Originator:***

No change from the generic procedures in clause 7.2.2.1.

If the Originator specifies a *missingData* condition with a duration value greater than the *periodicInterval* attribute of the <timeSeries> resource no notification on missing data points will be generated.

***Receiver:***

The following are additional Hosting CSE procedures to the generic resource handling procedures (Figure 7.2.2.2‑1 in clause 7.2.2.2). The additional procedures shall be inserted from Recv-6.2 to Recv-6.5 as below.

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 a ***Response Status Code*** indicating an "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 subscribed-to resource, addressed in the ***To*** parameter in the Request, is subscribable. Subscribable resource types are defined in TS-0001 [6]; 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 a "TARGET\_NOT\_SUBSCRIBABLE" error instead of the ***Response Status Code*** "INVALID\_CHILD\_RESOURCE\_TYPE".
2. Check if the *notificationEventType* is set to "Blocking\_Update".
* If the subscribed-to resource already has a subscription with this *notificationEventType* the Hosting CSE shall return the response primitive with an ***Response Status Code*** indicating a "BLOCKING\_SUBSCRIPTION\_ALREADY\_EXISTS" error if more than one notification of this type could be sent.
* If there is more than one *notificationURI* specified, the Hosting CSE shall return the response primitive with a ***Response Status Code*** indicating a "BAD\_REQUEST" error.
* If any resource-specific attributes of the <subscription> resource other than *eventNotificationCriteria* or *notificationURI* are specified the Hosting CSE shall return the primitive with a ***Response Status Code*** indicating a "BAD\_REQUEST" error.
* If any condition tag of the *eventNotificationCriteria* attribute other than *attribute* condition tag is specified, the Hosting CSE shall return the response primitive with a ***Response Status Code*** indicating a "BAD\_REQUEST" error.
1. Check if the *notificationEventType* is set to “Report on missing data points”. If the *missingData* attribute is not provided as well, the request shall be rejected with a ***Response Status Code*** indicating a "BAD\_REQUEST" error***.***
2. Check if the *missingData* element of eventNotificationCriteria is provided.
* If the subscribed-to resource (i.e. the resource given by the ***To*** parameter in the Request) is not a <timeSeries>, the request shall be rejected with a ***Response Status Code*** indicating a "BAD\_REQUEST" error.
1. 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.

a) 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 a "SUBSCRIPTION\_VERIFICATION\_INITIATION\_FAILED" error.

b) 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 a "SUBSCRIPTION\_VERIFICATION\_INITIATION\_FAILED" error.

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 ***Response Status Code*** indicating a “BAD\_REQUEST” error***.***
2. If the Originator provides *missingData*, check that subscribed-to resource is of type <timeSeries>. If not, the request shall be rejected with a ***Response Status Code*** indicating a "BAD\_REQUEST" error*.*
3. If both the *notificationEventType* and *operationMonitor* are present in the Request, the request shall be rejected with a ***Response Status Code*** indicating a "BAD\_REQUEST" error***.***
4. If the *notificationContentType* is invalid for a given operation (refer to oneM2M TS-0001 [6] Table 9.6.8-4: Default and allowed values of *notificationContentType*) the request shall be rejected with a ***Response Status Code*** indicating a "BAD\_REQUEST" error***.***

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

1. If the Originator does not provide *notificationContentType*, the Hosting CSE shall set it according to the default shown in oneM2M TS-0001 [6] Table 9.6.8-4: Default and allowed values of *notificationContentType.*
2. If the *notificationURI* is not the Originator, the Hosting CSE shall set the Originator’s ID as the <subscription> resource's *creator* attribute.
3. 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.
4. If the *notificationStatsEnable* attribute is set to true, the Hosting CSE shall start recording notification statistics in the *notificationStatsInfo* attribute once the <subscription> resource is created.

-------------------------------------------------- End of Change 8---------------------------------------

## ----------------------- Start of Change 9--------------------------------------------

#####

##### 7.5.1.2.4 Notification of Subscription Deletion

***Originator:***

When the <subscription> resource is deleted the Originator shall send Notify request primitives to the <subscription> resource's subscriberURI and *associatedCrossResourceSub* if they are configured:

a) subscriptionDeletion element of the notification data object set as true.

b) subscriptionReference element of the notification data object set as the resource identifier of the <subscription> resource.

c) The ***To*** parameter shall be set to the entity indicated in *subscriberURI* or *associatedCrossResourceSub*.

When only a target in *associatedCrossResourceSub* attribute is deleted, the Originator shall send a Notify request primitive to that target as follows:

a) subscriptionDeletion element of the notification data object set as true.

b) subscriptionReference element of the notification data object set as the resource identifier of the <subscription> resource.

c) The ***To*** parameter shall be set to the target removed from *associatedCrossResourceSub*

***Receiver:***

When a Hosting CSE receives a subscription deletion notification targeted to an existing <crossResourceSubscription> resource then the <crossResourceSubscription> resource is deleted using the procedure defined in clause 7.4.58.2.4 except when the crossResourceSubscription resource delete procedure is already in process to avoid a loop.

-------------------------------------------------- End of Change 9---------------------------------------

## ----------------------- Start of Change 10--------------------------------------------

Table 8.2.3‑6: Resource attribute short names (6/6)

| Attribute Name | Occurs in | Short Name |
| --- | --- | --- |
| *direction* | allJoynApp | ***dir*** |
| *objectPath* | allJoynSvcObject | ***ajop*** |
| *interfaceIntrospectXmlRef* | allJoynInterface | ***ajir*** |
| *input* | allJoynMethodCall | ***inp*** |
| *callStatus* | allJoynMethodCall | ***clst*** |
| *output* | allJoynMethodCall | ***out*** |
| *currentValue* | allJoynProperty | ***crv*** |
| *requestedValue* | allJoynProperty | ***rqv*** |
| *decision* | authorizationDecision | ***dec*** |
| *status* | authorizationDecision, authorizationPolicy, authorizationInformation, m2mServiceSubscriptionProfile | ***sus*** |
| *to* | authorizationDecision, authorizationPolicy | ***to\**** |
| *from* | authorizationDecision, authorizationInformation | ***fr\**** |
| *requestedResourceType* | authorizationDecision | ***rrt*** |
| *operation* | authorizationDecision | ***op\**** |
| *filterUsage* | authorizationDecision | ***fu*** |
| *roleIDs* | authorizationDecision, authorizationInformation | ***rids\**** |
| *tokenIDs* | authorizationDecision, authorizationInformation | ***tids\**** |
| *tokens* | authorizationDecision | ***tkns\**** |
| *requestTime* | authorizationDecision | ***rtm*** |
| *originatorLocation* | authorizationDecision | ***olo*** |
| *originatorIP* | authorizationDecision | ***oip*** |
| *policies* | authorizationPolicy | ***ps*** |
| *combiningAlgorithm* | authorizationPolicy | ***ca*** |
| *ontologyFormat* | ontology | ***ontf*** |
| *ontologyContent* | ontology | ***ontc*** |
| *sourceOntology* | ontologyMapping | ***sont*** |
| *targetOntology* | ontologyMapping | ***tont*** |
| *mappingPolicy* | ontologyMapping | ***mpol*** |
| *mappingAlgorithmLinks* | ontologyMapping | ***mpal*** |
| *mappingResultFormat* | ontologyMapping | ***mprf*** |
| *mappingResult* | ontologyMapping | ***mpr*** |
| *executable* | ontologyMappingAlgorithm | ***exec*** |
| *algorithmType* | ontologyMappingAlgorithm | ***algt*** |
| *mappingThreshold* | ontologyMappingAlgorithm | ***mpth*** |
| *memberFilter* | semanticMashupJobProfile | ***mbft*** |
| *smiID* | semanticMashupJobProfile | ***miid*** |
| *inputDescriptor* | semanticMashupJobProfile | ***iptd*** |
| *outputDescriptor* | semanticMashupJobProfile | ***uptd*** |
| *functionDescriptor* | semanticMashupJobProfile | ***fucd*** |
| *smjpID* | semanticMashupInstance | ***mjid*** |
| *smjpInputParameter* | semanticMashupInstance, semanticMashupResult | ***jpin*** |
| *memberStoreType* | semanticMashupInstance | ***mst*** |
| *mashupMember* | semanticMashupInstance | ***msm*** |
| *resultGenType* | semanticMashupInstance | ***rgt*** |
| *periodForResultGen* | semanticMashupInstance | ***prg*** |
| *mashupResultFormat* | semanticMashupResult | ***mrf*** |
| *mashupResult* | semanticMashupResult | ***mrt*** |
| *ruleRepresentation* | reasoningRules | ***rrep*** |
| *ruleRepresentationFormat* | reasoningRules | ***rrepf*** |
| *reasoningType* | reasoningJobInstance | ***rtyp*** |
| *reasoningMode* | reasoningJobInstance | ***rmod*** |
| *reasoningPeriod* | reasoningJobInstance | ***rper*** |
| *factSet* | reasoningJobInstance | ***rfst*** |
| *ruleSet* | reasoningJobInstance | ***rrst*** |
| *resultRepresentation* | reasoningJobInstance | ***rsrp*** |
| *resultRepresentationFormat* | reasoningJobInstance | ***rsrpf*** |
| *numberImpactedCSEs* | AEContactList | ***nic*** |
| *externalGroupID* | LocalMulticastGroup, remoteCSE, nwMonitoringReq | ***egid*** |
| *multicastAddress* | LocalMulticastGroup | ***mad*** |
| *multicastGroupFanoutTarget* | LocalMulticastGroup | ***mgft*** |
| *memberList* | LocalMulticastGroup | ***mli*** |
| *responseTarget* | LocalMulticastGroup | ***rstt*** |
| *responseTimeWindow* | LocalMulticastGroup | ***rstw*** |
| *TMGI* | LocalMulticastGroup | ***tmgi*** |
| *sessionOriginatorID* | multimediaSession | ***soi*** |
| *acceptedSessionDescriptions* | multimediaSession | ***asd*** |
| *offeredSessionDescriptions* | multimediaSession | ***osd*** |
| *sessionState* | multimediaSession | ***sst*** |
| *triggerPurpose* | triggerRequest | ***tpe*** |
| *triggerStatus* | triggerRequest | ***tst*** |
| *triggerValidityTime* | triggerRequest | ***tvt*** |
| *triggerInfoAE-ID* | triggerRequest | ***tiae*** |
| *triggerInfoAddress* | triggerRequest | ***tia*** |
| *triggerInfoOperation* | triggerRequest | ***tio***  |
| *targetedResourceType* | triggerRequest | ***tirt***  |
| *triggerReference* | triggerRequest | ***trf*** |
| *regularResourcesAsTarget* | crossResourceSubscription | ***rrat*** |
| *subscriptionResourcesAsTarget* | crossResourceSubscription | ***srat*** |
| *regularResourcesAsTargetSubscriptions* | crossResourceSubscription | ***rrats*** |
| *timeWindowType* | crossResourceSubscription | ***twt*** |
| *timeWindowSize* | crossResourceSubscription | ***tws*** |
| *eventNotificationCriteriaSet* | crossResourceSubscription | ***encs*** |
| *associatedCrossResourceSub* | subscription | ***acrs*** |
| *volumePerNode* | backgroundDataTransfer | ***vpn*** |
| *numberOfNodes* | backgroundDataTransfer | ***non*** |
| *desiredTimeWindow* | backgroundDataTransfer | ***dtw*** |
| *transferSelectionGuidance* | backgroundDataTransfer | ***tsg*** |
| *geographicInformation* | backgroundDataTransfer | ***ggi*** |
| *groupLink* | backgroundDataTransfer | ***gli*** |
| *transactionLockTime* | transactionMgmt, transaction | ***tltm*** |
| *transactionExecuteTime* | transactionMgmt, transaction | ***text*** |
| *transactionCommitTime* | transactionMgmt, transaction | ***tct*** |
| *transactionExpirationTime* | transactionMgmt | ***tept*** |
| *transactionMode* | transactionMgmt | ***tmd*** |
| *transactionLockType* | transactionMgmt, transaction | ***tltp*** |
| *transactionControl* | transactionMgmt, transaction | ***tctl*** |
| *transactionState* | transactionMgmt, transaction | ***trst*** |
| *transactionMaxRetries* | transactionMgmt | ***tmr*** |
| *transactionMgmtHandling* | transactionMgmt | ***tmh*** |
| *requestPrimitives* | transactionMgmt | ***rqps*** |
| *responsePrimitives* | transactionMgmt | ***rsps*** |
| *transactionID* | transaction | ***tid*** |
| *sessionEndpoints* | e2eQosSession | ***eqse*** |
| *e2eQosRequirements* | e2eQosSession | ***eqsr*** |
| *e2eQosPolicies* | e2eQosSession | ***eqsp*** |
| *e2eQosStatus* | e2eQosSession | ***eqss*** |
| *beaconRequester* | timeSyncBeacon | ***bcnr*** |
| *beaconCriteria* | timeSyncBeacon | ***bcnc*** |
| *beaconInterval* | timeSyncBeacon | ***bcni*** |
| *beaconThreshold* | timeSyncBeacon | ***bcnt*** |
| *beaconNotificationURIs* | timeSyncBeacon | ***bcnu*** |
| *monitorEnable* | nwMonitoringReq | ***mnen*** |
| *monitorStatus* | nwMonitoringReq | ***mnst*** |
| *failureReason* | nwMonitoringReq | ***frsn*** |
| *geographicArea* | nwMonitoringReq | ***geoa*** |
| *congestionLevel* | nwMonitoringReq | ***cgnl*** |
| *congestionStatus* | nwMonitoringReq | ***cgns*** |
| *numberOfDevices* | nwMonitoringReq | ***ndev*** |
| *notificationStatsEnable* | subscription, crossResourceSubscription | ***nse*** |
| *notificationStatsInfo* | subscription, crossResourceSubscription | ***nsi*** |
| *IDList* | primitiveProfile | ***idl*** |
| *resourceTypes* | primitiveProfile | ***rtys*** |
| *resourceIDs* | primitiveProfile | ***rsds*** |
| *releaseVersions* | primitiveProfile | ***rvs*** |
| *additions* | primitiveProfile | ***adds*** |
| *deletions* | primitiveProfile | ***dels*** |
| *applicability* | primitiveProfile | ***appl*** |
| *processStatus* | processManagement | ***prst*** |
| *processControl* | processManagement | ***prct*** |
| *currentState* | processManagement | ***cust*** |
| *activateConditions* | processManagement | ***atcos*** |
| *endConditions* | processManagement | ***encos*** |
| *initialState* | processManagement | ***inst*** |
| *stateActive* | state | ***sact*** |
| *stateAction* | state | ***stac*** |
| *stateTransitions* | state | ***sttrs*** |
| *actionPriority* | action | ***apy*** |
| *subjectResourceID* | action | ***sri*** |
| *evalCriteria* | action, dependency | ***evc*** |
| *evalMode* | action | ***evm*** |
| *evalControlParam* | action | ***ecp*** |
| *dependencies* | action | ***dep*** |
| *objectResourceID* | action | ***orc*** |
| *actionPrimitive* | action | ***apv*** |
| *input* | action | ***ipu*** |
| *actionResult* | action | ***air*** |
| *sufficient* | dependency | ***sfc*** |
| *referencedResourceID* | dependency | ***rri*** |
| *M2M-Sub-ID* | m2mServiceSubscriptionProfile | ***msui*** |
| *M2M-SS-ID* | m2mServiceSubscriptionProfile | ***mssi*** |
| *M2M-User-ID* | serviceSubscribedUserProfile | ***mui*** |
| *allowedUsers* | serviceSubscribedAppRule | ***allu*** |
| *activationTime* | m2mServiceSubscriptionProfile | ***actt*** |
| *deactivationTime* | m2mServiceSubscriptionProfile | ***dact*** |
| *serviceSubscriptionDuration* | m2mServiceSubscriptionProfile | ***ssd*** |
| *currentNumAEs* | m2mServiceSubscriptionProfile | ***cna*** |
| *currentNumNodes* | m2mServiceSubscriptionProfile | ***cnn*** |
| *maxNumAEs* | m2mServiceSubscriptionProfile, serviceSubscribedUserProfile | ***mna*** |
| *maxNumNodes* | m2mServiceSubscriptionProfile, serviceSubscribedUserProfile | ***mnns*** |
| *maxNumBytes* | m2mServiceSubscriptionProfile, serviceSubscribedUserProfile | ***mnb*** |
| *maxRequestRate* | m2mServiceSubscriptionProfile, serviceSubscribedUserProfile | ***mrr*** |
| *maxNumContainers* | m2mServiceSubscriptionProfile, serviceSubscribedUserProfile | ***mnc*** |
| *maxNumInstancesPerContainer* | m2mServiceSubscriptionProfile, serviceSubscribedUserProfile | ***mnis*** |
| *maxNumTimeSeries* | m2mServiceSubscriptionProfile, serviceSubscribedUserProfile | ***mnt*** |
| *maxNumInstancesPerTimeSeries* | m2mServiceSubscriptionProfile, serviceSubscribedUserProfile | ***mnit*** |
| *maxMembersPerGroup* | m2mServiceSubscriptionProfile, serviceSubscribedUserProfile | ***mmg*** |
| *maxNotificationRate* | m2mServiceSubscriptionProfile, serviceSubscribedUserProfile | ***mnr*** |
| *maxNumFlexContainers* | m2mServiceSubscriptionProfile, serviceSubscribedUserProfile | ***mnf*** |
| *maxNumInstancesPerFlexContainer* | m2mServiceSubscriptionProfile, serviceSubscribedUserProfile | ***mnif*** |
| *defaultAccessControlPrivileges* | m2mServiceSubscriptionProfile, serviceSubscribedUserProfile | ***dacp*** |
| *campaignEnabled* | softwareCampaign | ***swce*** |
| *campaignStatus* | softwareCampaign | ***swcs*** |
| *softwareTargets* | softwareCampaign | ***swts*** |
| *softwareTriggerCriteria* | softwareCampaign | ***swtc*** |
| *softwareOperation* | softwareCampaign | ***swop*** |
| *softwareLinks* | softwareCampaign | ***swlk*** |
| *aggregatedSoftwareStatus* | softwareCampaign | ***asws*** |
| *individualSoftwareStatuses* | softwareCampaign | ***isws*** |
| NOTE: \* marked short names have been already assigned in Table 8.2.2-1. |

-------------------------------------------------- End of Change 9---------------------------------------