|  |
| --- |
|  |

|  |
| --- |
| CHANGE REQUEST |
| Meeting ID:\* | SDS 49 |
| Source:\* | Miguel Angel Reina Ortega, ETSI, MiguelAngel.ReinaOrtega@etsi.org Neeta Meshram, C-DOT, neeta@cdot.inBob Flynn, Exacta GSS, bob.flynn@exactagss.com  |
| Date:\* | 2021-02-03 |
| Reason for Change/s:\* | notificationEventType for timeSeries |
| 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-2020-0361R04-TS-0001\_notificationEventType\_for\_timeSeries\_R3[ ]  STE Small Technical Enhancements / < Work Item number (optional)>Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0001 v4.8.0 |
| Clauses \* | 9.6.8, 10.2.4.29 |
| Type of change: \* | [ ]  Editorial change[ ]  Bug Fix or Correction[x]  Change to existing feature or functionality[ ]  New feature or functionalityOnly ONE of the above shall be ticked |
| Other TS/TR(s) impacted |  TS-0004 |
| 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

This CR proposes a new notificationEventType which will be used for timeSeries. This will allow:

* Configure properly the subscription to the timeSeries to get notifications on the number of missingDataPoints
* Notifications generated from timeSeries procedure to be clearly distinguished

This CR also provides an update to the Create and Update subscription procedures to check presence of missingData when notificationEventType is provided and has “Report on generated missing data points” as a value.

And finally, the CR proposes a new value for notificationContentType specific for timeSeries notifications.

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

**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..7 | 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.
8. Report on generated missing data points. The *notificationContentType* shall be “TimeSeries notification”.

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" or “Deletion 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 or deletion shall result in a notification. If ANY resource type specified on this list is created or deleted, then a notification shall be generated. If a resource type that is not specified in this list is created or deleted, 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>*. This condition only applies when *notificationEventType* has a value of “Report of generated missing data points”. 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/XOR) to be used for the condition tags *createdBefore, createdAfter, modifiedSince, unmodifiedSince, stateTagSmaller, stateTagBigger, expireBefore, expireAfter, sizeAbove, sizeBelow*. The default value is logical AND. |

The rules when multiple conditions are used together shall be as follows:

Different condition tags shall use the "AND/OR/XOR" logical operation based on the *filterOperation* specified;

Same condition tags shall use the "OR" logical operation.

The XOR operation evaluates to true if and only if an odd number of its inputs are true.

No mixed AND/OR/XOR filter operation will be supported.

Table 9.6.8-4 defines the default and allowed values of *notificationContentType* for each of the supported values of *notificationEventType*.

**Table 9.6.8-4:** **Default and allowed values of *notificationContentType***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **notificationEventType****notificationContentType** | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** |
| "modified attributes"; | valid | n/a | n/a | n/a | n/a | n/a | valid (default) | n/a |
| "all attributes"; | valid (default) | valid (default) | valid (default) | valid (default) | valid (default) | n/a | n/a | n/a |
| "ID" of the resource indicated in the *notificationEventType* condition. | valid | valid | valid | valid | valid | n/a | n/a | n/a |
| “Trigger Payload” | n/a | n/a | n/a | n/a | n/a | valid (default) | n/a | n/a |
| “TimeSeries notification” | n/a | n/a | n/a | n/a | n/a | n/a | n/a | valid (default) |

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

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

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 shall request the creation of a *<subscription>* resource and set the *missingData* in the *eventNotificationCriteria* conditions to specify the reporting policy. The *notificationEventType* element in the *eventNotificationCriteria* shall have a value of “Report on generated missing data points”. 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 “list of missing data” and "number of missing data" 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 2---------------------------------------

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

10.2.10.2 Create *<subscription>*

This procedure shall be used to request the creation of a new *<subscription>* resource to instruct the Hosting CSE to send notifications to configured Subscriber(s) for modifications of a subscribed-to resource. The generic create procedure is described in clause 10.1.2.

**Table 10.2.10.2-1: *<subscription>* CREATE**

| ***<subscription>* CREATE**  |
| --- |
| Information in Request message | All parameters defined in table 8.1.2-3 apply with the specific details for:***Content*:** The resource content shall provide the information as defined in clause 9.6.8 |
| Processing at Originator before sending Request | According to clause 10.1.2 with the following additions:The Request shall address a subscribable resourceThe Request shall include a <*subscription>* resource representation withtheattribute *notificationURI*If the *notificationURI* attribute includes Notification Target(s) which is/are not targeting the Originator, the Originator should send the request as non-blocking request (see clauses 8.2.2 and 9.6.12) |
| Processing at Receiver | According to clause 10.1.2 with the following additions:The Hosting CSE shall validate the followings:* Check if the subscribed-to resource, addressed in the ***To*** parameter in the Request, is a subscribable resourceCheck if the Originator has privileges for retrieving the subscribed-to resourceIn case a <*subscription*> resource representation is provided with a *notificationEventType* tag equal to “Update to attributes of the subscribed-to resource with blocking of the triggering UPDATE operation” in the *eventNotificationCriteria* attribute, check that no other subscriptions with this setting exist for the resource in the ***To*** parameter, check that only one entity is targeted by the *notificationURI* attribute and check that this entity has privileges for updating the subscribed-to resource.
* If the <subscription> resource representation is provided with *notificationEventType* tag equal to "Report on generated missing data points", check that *missingData* attribute is provided.
* If missingData attribute is provided, check that subscribed-to resource is <timeSeries> resource.If an entity listed in thenotificationURI is not the Originator, the Hosting CSE may send a Notify request to that entity to verify this *<subscription>* creation request. If the Hosting CSE initiates the verification, it shall check if the verification result in the Notify response is successful or not. If any of the entities listed in the *notificationURI* attribute fails verification then the *<subscription>* create process fails.If the <*subscription*> resource representation includes a *primitiveProfileID* attribute configured with a value and the Hosting CSE does not support primitive profile functionality, then the <*subscription*> resource create process fails.

If any of the checks above fails, the Hosting CSE shall send an unsuccessful response to the Originator with corresponding error information. Otherwise, the Hosting CSE shall create the *<subscription>* resource and send a successful response to the Originator. Upon successful creation of a <*subscription*> resource, the Hosing CSE shall evaluate subsequent operations on the subscribed-to resource and trigger notifications in line with the notification policies provisioned in the created <*subscription*> resource. If the *notifStatType* attribute is configured, the Hosting CSE shall collect and record notification statistics as defined in clause 10.2.10.27. |
| Information in Response message | All parameters defined in table 8.1.3-1 apply with the specific details for:***Content*:** address of the created *<subscription>* resource, according to clause 10.1.2 |
| Processing at Originator after receiving Response | According to clause 10.1.2 |
| Exceptions | According to clause 10.1.2 |

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

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

10.2.10.4 Update *<subscription>*

This procedure shall be used to update an existing subscription, e.g. extension of its lifetime or the modification of the list of Notification Targets provided in the *notificationURI* attribute. The generic update procedure is described in clause 10.1.4.

**Table 10.2.10.4-1: *<subscription>* UPDATE**

| ***<subscription>* UPDATE** |
| --- |
| Information in Request message | All parameters defined in table 8.1.2-3 apply with the specific details for:***Content:*** attributes of the *<subscription>* resource as defined in clause 9.6.8 which need be updated |
| Processing at Originator before sending Request | According to clause 10.1.4 |
| Processing at Receiver | According to clause 10.1.4 with the following additions:* If the *notificationURI* attribute contains Notification Target(s) that is/are not the Originator, see applicable processing in table 10.2.10.2-1 in clause 10.2.10.2
* If *notificationEventType* is provided as "Report on generated missing data points", check if *missingData is provided.*
* If *missingData* attribute is provided, check that subscribed-to resource is <timeSeries> resource.
* If the *latestNotify* attribute is set during this UPDATE operation, the Hosting CSE shall assign ***Event Category*** parameterof value'latest' of the notifications generated pertaining to the subscription created and remove all buffered pending notifications for this subscription except for the latest one.
* If the <*subscription*> resource representation includes a *primitiveProfileID* attribute configured with a value and the Hosting CSE does not support primitive profile functionality, then the <*subscription*> resource update process fails.

Upon successful updating of a <*subscription*> resource, the Hosing CSE shall evaluate subsequent operations on the subscribed-to resource and trigger notifications in line with the new notification policies provisioned in the created <*subscription*> resource. If the *notifStatType* attribute is configured, the Hosting CSE shall collect and record notification statistics as defined in clause 10.2.10.27. |
| Information in Response message | According to clause 10.1.4 |
| Processing at Originator after receiving Response | According to clause 10.1.4 |
| Exceptions | According to clause 10.1.4 |

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

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

**Table 9.6.8-2: Attributes of *<subscription>* resource**

| **Attributes of *<subscription>*** | **Multiplicity** | **RW/****RO/****WO** | **Description** |
| --- | --- | --- | --- |
| *resourceType* | 1 | RO | See clause 9.6.1.3. |
| *resourceID* | 1 | RO | See clause 9.6.1.3. |
| *resourceName* | 1 | WO | See clause 9.6.1.3. |
| *parentID* | 1 | RO | See clause 9.6.1.3. |
| *expirationTime* | 1 | RW | See clause 9.6.1.3. |
| *creationTime* | 1 | RO | See clause 9.6.1.3. |
| *lastModifiedTime* | 1 | RO | See clause 9.6.1.3. |
| *labels* | 0..1 (L) | RW | See clause 9.6.1.3. |
| *accessControlPolicyIDs* | 0..1 (L) | RW | See clause 9.6.1.3. |
| *dynamicAuthorizationConsultationIDs* | 0..1 (L) | RW | See clause 9.6.1.3. |
| *creator* | 0..1 | WO | See clause 9.6.1.3. |
| *holder* | 0..1 | RW | See clause 9.6.1.3 |
| *eventNotificationCriteria* | 0..1 | RW | This attribute (notification policy) indicates the event criteria for which a notification is to be generated. When no *eventNotificationCriteria* attribute is present in a <*subscription*> resource, the Hosting CSE shall trigger notifications for this subscription when any of the attributes of the subscribed-to resource is modified. |
| *expirationCounter* | 0..1 | RW | This attribute (notification policy) indicates that the subscriber wants to set the life of this subscription to a limit of a maximum number of notifications. When the number of notifications sent reaches the count of this counter, the *<subscription>* resource shall be deleted, regardless of any other policy. |
| *notificationURI* | 1 (L) | RW | This attribute shall be configured as a list consisting of one or more targets that the Hosting CSE shall send notifications to. A target shall be formatted as a oneM2M compliant Resource-ID as defined in clause 7.2 or as an identifier compliant with a oneM2M supported protocol binding (e.g. http, coap, mqtt). If a target is formatted as a oneM2M compliant Resource-ID, then the target shall be formatted as a structured or unstructured CSE-Relative-Resource-ID, SP-Relative-Resource-ID, and/or Absolute-Resource-ID of an <*AE*> or <CSEBase> resource. A Hosting CSE shall use this information to determine proper pointOfAccess, requestReqchability and/or pollingChannel information needed to send a notification to the target. The following is an example./CSE0001/AE0001For a target that is formatted as an identifier compliant with a oneM2M supported protocol binding, the details of this format are defined by the respective oneM2M protocol specification. The following is an example of an HTTP URI compliant with oneM2M HTTP protocol binding.https://172.25.30.25:7000/notification/handlerFor a subscription to a <fanoutpoint> resource, if <subscription> resource in request contains a notificationForwardingURI, then the group hosting CSE shall configure the *notificationURI* of the fanout subscription request with an address specified by the Group Hosting CSE that can be used by the Group Hosting CSE to receive aggregated notifications. A notification serialization type may be appended to each notification target configured in this list. The Hosting CSE shall serialize notifications and send it to a notification target based on this serialization type indicator. Possible serialization types are defined in the TS-0004 [3] (e.g. XML, JSON or CBOR). If a notification serialization type is not appended to a notification target, a default shall apply based on the Hosting CSE local policy. The syntax for appending a serializatino type to a notification target shall use the “?” delimiter character as shown in the below examples.http://mydomain/notificationHandler?ct=jsonCSE02/base/ae2?ct=xml |
| *groupID* | 0..1 | RW | The ID of a *<group>* resource in case the subscription is made through a group. This attribute may be used in the ***Filter Criteria*** to discover all subscription resources created via a <fanOutPoint> resource to a specific groupID. |
| *notificationForwardingURI* | 0..1(L) | RW | The attribute shall be present only for group related subscriptions. If the subscriber intends the Group Hosting CSE to aggregate the notifications, the attribute shall be set identical to the *notificationURI* attribute. It shall be used by Group Hosting CSE for forwarding aggregated notifications. See clauses 10.2.7.10 and 10.2.7.11. |
| *batchNotify* | 0..1 | RW | This attribute (notification policy) indicates that the subscription originator wants to receive batches of notifications rather than receiving them one at a time. This attribute includes: the number of notifications to be batched for delivery and the duration. When only the number is specified by the subscription originator, the Hosting CSE shall set the default duration given by M2M Service Provider. If *batchNotify* is used simultaneously with *latestNotify*, only the latest notification shall be sent and have the ***Event Category*** set to "latest". |
| *rateLimit* | 0..1 | RW | This attribute (notification policy) indicates that the subscriber wants to limit the rate at which it receives notifications. This attribute expresses the subscriber's notification policy and includes two values: a maximum number of events that may be sent within some duration, and the *rateLimit* window duration. When the number of generated notifications within the *rateLimit* window duration exceeds the maximum number, notification events are temporarily stored, until the end of the window duration, when the sending of notification events restarts in the next window duration. The sending of notification events continues as long as the maximum number of notification events is not exceeded during the window duration. The *rateLimit* policy may be used simultaneously with other notification policies. |
| *preSubscriptionNotify* | 0..1 | WO | This attribute (notification policy) indicates that the subscriber wants to be sent notifications for events that were generated prior to the creation of this subscription. This attribute has a value of the number of prior notification events requested. If up-to-date caching of retained events is supported on the Hosting CSE and contains the subscribed events, then prior notification events will be sent up to the number requested. The *preSubscriptionNotify* policy may be used simultaneously with any other notification policy. |
| *pendingNotification* | 0..1 | RW | This attribute (notification policy), if set, indicates how missed notifications due to a period of no connectivity are handled (according to the reachability and notification schedules). The possible values for *pendingNotification are*:"sendLatest";"sendAllPending".This policy depends upon caching of retained notifications on the hosted CSE. When this attribute is set to "sendLatest", only the last notification shall be sent and it shall have the ***Event Category*** set to "latest". If this attribute is not present, the Hosting CSE sends no missed notifications. This policy applies to all notifications regardless of the selected delivery policy (*batchNotify*, *latestNotify*, etc.) Note that unreachability due to reasons other than scheduling is not covered by this policy. |
| *notificationStoragePriority* | 0..1 | RW | Indicates that the subscriber wants to set a priority for this subscription relative to other subscriptions belonging to this same subscriber. This attribute sets a number within the priority range. When storage of notifications exceeds the allocated size, this policy is used as an input with the storage congestion policy (*notificationCongestionPolicy*) specified in clause 9.6.3 to determine which stored and generated notifications to drop and which ones to retain. |
| *latestNotify* | 0..1 | RW | This attribute (notification policy) indicates if the subscriber wants only the latest notification. If multiple notifications of this subscription are buffered, and if the value of this attribute is set to true, then only the last notification shall be sent and it shall have the ***Event Category*** value set to "latest". |
| *notificationContentType* | 1 | RW | Indicates a notification content type that shall be contained in notifications. The allowed values are:"modified attributes";"all attributes";"ID" of the resource indicated in the *notificationEventType* condition.Trigger Payload;“TimeSeries notification”For a list of the default and allowed values of *notificationContentType* for each of the supported values of *notificationEventType* refer to Table 9.6.8-4. |
| *notificationEventCat* | 0..1 | RW | This attribute (notification policy) indicates the subscriber's requested ***Event Category*** to be used for notification messages generated by this subscription. |
| *subscriberURI* | 0..1 | WO | This attribute shall be configured with the target of the subscriber. The target is used by the Hosting CSE to determine where to send a notification when the subscription is deleted. A target shall be formatted as a oneM2M compliant Resource-ID as defined in clause 7.2 or as an identifier compliant with one of the oneM2M supported protocol bindings (the detailed format of which are defined by each respective oneM2M protocol binding specification). |
| *associatedCrossResourceSub* | 0..1 | RW | This attribute lists *the identifier of <crossResourceSubscription>* resources where this *<subscription>* is involved in.  |
| *primitiveProfileID* | 0..1 | RW | This attribute lists the identifier of a *<primitiveProfile>* resource that specifies attributes and parameters to be added, removed, or modified in the notifications for this subscription.  |
| *notifStatType* | 0..1(L) | RW | Indicates a list of the types of notification statistics that the Hosting CSE shall collect for each notification target specified by the *notificationURI* attribute of this <*subscription*> resource. When this attribute is configured with a value other than NULL upon creation or update of this resource, the Hosting CSE shall set the value of the *notifStatInfo* attribute and begin collecting notification statistics for each notification generated for this resource clearing any statistics that were previously stored in the *notifStatInfo* attribute. When this attribute is updated with a value of NULL, the Hosting CSE shall delete this attribute and stop collecting notification statistics for this <*subscription*> resource, however the Hosting CSE shall not delete the *notifStatInfo* attribute.The allowed types of notification statistics are:* Total number of notification requests sent to a notification target
* Total number of notification responses received from a notification target
 |
| *notifStatInfo* | 0..1(L) | RO | A list containing notification statistics specified by the *notifStatType* attribute. For each notification target specified by the *notificationURI* attribute, the Hosting CSE shall maintain a separate set of notification statistics within this list.Refer to oneM2M TS 0004 [3] for further details regarding the format of this attribute. |

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