|  |
| --- |
|  |

|  |
| --- |
| CHANGE REQUEST |
| Meeting ID:\* |  SDS #60 |
| Source:\* | Andreas Kraft, DT, Andreas.Kraft@t-systems.com Andreas Neubacher, DT, Andreas.Neubacher@magenta.at Miguel Angel Reina Ortega, ETSI, MiguelAngel.ReinaOrtega@etsi.org |
| Date:\* | 2023-06-19 |
| Reason for Change/s:\* | Set “creator” attribute to request originator in notifications (TS-0004) |
| CR against: Release\* | Release 5 |
| CR against: WI\* | [ ]  Active WI-xxxx[ ]  MNT maintenance / < Work Item number(optional)>Is this a mirror CR? Yes [ ]  No [ ] mirror CR number: (Note to Rapporteur - use latest agreed revision)[x]  STE Small Technical Enhancements / < Work Item number (optional)>Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0004 v.4.15.0 |
| Clauses \* | 7.5.1.2.2 |
| 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 |
| Impacted other TS/TR(s) |  |
| Post Freeze checking:\* | This CR contains only essential changes and corrections? YES [x]  NO [ ] This CR may break backwards compatibility with the last approved version of the TS? YES [ ]  NO [x]  |
| Template Version: January 2017 (Do not modify) |

**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

GUIDELINES for Change Requests:

Provide an informative introduction containing the problem(s) being solved, and a summary list of proposals.

Each CR should contain changes related to only one particular issue/problem.

In case of a correction, and the change apply to previous releases, a separate “mirror CR” should be posted at the same time of this CR

Mirror CR: applies only when the text, including clause numbering are exactly the same.

Companion CR: applies when the change means the same but the baselines differ in some way (e.g. clause number).

Follow the principle of completeness, where all changes related to the issue or problem within a deliverable are simultaneously proposed to be made E.g. A change impacting 5 tables should not only include a proposal to change only 3 tables. Includes any changes to references, definitions, and acronyms in the same deliverable.

Follow the drafting rules.

All pictures must be editable.

Check spelling and grammar to the extent practicable.

Use Change bars for modifications.

The change should include the current and surrounding clauses to clearly show where a change is located and to provide technical context of the proposed change. Additions of complete clauses need not show surrounding clauses as long as the proposed clause number clearly shows where the new clause is proposed to be located.

Multiple changes in a single CR shall be clearly separated by horizontal lines with embedded text such as, start of change 1, end of change 1, start of new clause, end of new clause.

When subsequent changes are made to content of a CR, then the accepted version should not show changes over changes. The accepted version of the CR should only show changes relative to the baseline approved text.

Introduction

This Change Request proposes to set the *creator* attribute to the originator of the request that caused the event that led to the notification. However, this is only be done if there is a request that caused the event. For example, a subscription that monitors a <TimeSeries> resource for missing data point raises events without a direct request. In this case the notification shall not set the *creator* attribute in the notification.

The use case for this change request is, for example, an AE (R) creates and monitors a <container> resource and gives multiple other AEs (S) CREATE permissions to it in order to write data, for example commands. Currently, there is no native way to determine, which of the other AEs created a <contentInstance>, for example with a command, besides providing the information either in the content or as a *label*, which both may cause interoperability problems.

With the proposed change the first AE (R) will now receive this information in the notification request.

This change request is for R5, if accepted then further Change Requests for R2, R3, and R4 will be submitted.

Change 1: Added adding the originator to the (already defined) notification *creator* attribute.

**R01**

After discussing the CR during TP#60 it was decided to make this added feature (ie. adding the originator of a request that causes a subscription notification to the notification) optional. This will keep backward compatibility as well as protect the identity of originator by default.

See also the TS-0001 CR SDS-2023-0143 that introduces a new attribute to the <subscription> resource type to switch on the new behaviour.

Change 1: Add additional text to Change 1 to reflect the handling of the *enableEventNotificationOriginator* attribute.

Change 2: Add the *enableEventNotificationOriginator* attribute to <subscription>.

Change 3: Add the short name “eeno” for *enableEventNotificationOriginator*.

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

##### 7.5.1.2.2 Notification for <subscription> resources

When the notification message is forwarded or aggregated by transit CSEs, the Originator or a transit CSE shall check whether there are notification policies to enforce between subscription resource Hosting CSE and the notification target. In that case, the transit CSE as well as the Originator shall process Notify request primitive(s) by using the corresponding policy and send processed Notify request primitive(s) to the next CSE with notification policies related to the enforcement so that the transit CSE is able to enforce the policy defined by the subscriber. The notification policies related to the enforcement at this time is verified by using the subscription reference in the Notify request primitive. In the notification policies, the *latestNotify* attribute is only enforced in the transit CSE as well as the Originator.

If ***Event Category*** parameter is set to "latest" in the notification request primitive, the transit CSE as well as Originator shall cache the most recent Notify request. That is, if a new Notify request is received by the CSE with a subscription reference that has already been buffered for a pending Notify request, the newer Notify request will replace the buffered older Notify request.

***Originator:***

When an event is generated, the Originator shall execute the following steps in order:

Step 1.0 Check the *eventNotificationCriteria* attribute of the <subscription> resource associated with the modified resource:

* If the *eventNotificationCriteria* attribute is set, then the Originator shall check whether the corresponding event matches with the event criteria. If multiple matching conditions of different types (i.e. different condition tags) are present in the *eventNotificationCriteria* attribute, then the combined condition shall be derived by applying the logical operation specified by the *filterOperation* condition. By default the logical AND operation shall be used if the *filterOperation* condition is not present.
* If *notificationEventType* is not set within the *eventNotificationCriteria* attribute and the *operationMonitor* is also not present, the Originator shall use the default setting of "Update\_of\_Resource" to compare against the event.
* If the *notificationEventType* has the value "Create\_of\_Direct\_Child\_Resource" or "Delete of Direct Child Resource" and the *childResourceType* condition is also present, then the matching event shall only be detected if one of the child resource types present in the list has been created or deleted, respectively. If the *childResourceType* condition is not present then a matching event is generated whenever any child resource is created or deleted.
* If the *notificationEventType* has either an explicit or default value of "Update\_of\_Resource" and the *attribute* condition is also present then the matching event shall only be detected if one of the attributes in the list has been updated. If the *attribute* condition is not present then a matching event is generated whenever any attribute has been updated.
* If the event matches, go to the step 2.0. Otherwise, the Originator shall discard the corresponding event.
* If the *eventNotificationCriteria* attribute is not configured, the Originator shall use the default setting of "Update\_of\_Resource" for the *notificationEventType* and then continue with the step 2.0.

Step 2.0 The Originator shall check the notification policy as described in the below steps, but the notification policy may be checked in different order. After checking the notification policy in step 2.0 (i.e. from step 2.1 to step 2.6), then continue with step 3.0.

Step 2.1 The Originator shall determine the type of the notification per the *notificationContentType* attribute. The possible values of for *notificationContentType* attribute are "Modified Attributes", "All Attributes", "ResourceID", "Trigger Payload" or “TimeSeries notification”. This attribute may be used jointly with the *notification*EventType attribute in the eventNotificationCriteria to determine if it is the attributes/resourceID of the subscribed-to resource or the attributes/resourceID of the child resource of the subscribed-to resource that shall be returned in the content of the notification:

* If the value of *notificationContentType* is set to "Modified Attributes", the Notify request primitive shall include the partial resource containing modified attribute(s) only (Refer to clause 7.2.1.2 for response content description).
* If the value of *notificationContentType* is set to "All Attributes", the Notify request primitive shall include the complete resource with all attributes (Refer to clause 7.2.1.2 for response content description).
* If the value of *notificationContentType* is set to "ResourceID", the Notify request primitive shall include the URI of the resource (Refer to clause 7.2.1.2 for response content description).
* If the value of *notificationContentType* is set to "Trigger Payload", the Notify request primitive shall include the trigger payload (Refer to clause 9.2.1 for trigger payload description).
* If the value of *notificationContentType* is set to "TimeSeries notification", the Notify request primitive shall include a timeSeriesNotification (Refer to clause 6.3.5.69 for timeSeriesNotification description).

In addition to the procedure described above, if the *primitiveProfileID* attribute of the <subscription> resource is configured with the resource identifier of a <primitiveProfile> resource, then the Originator shall apply the <primitiveProfile> resource to the request parameters of the Notify request primitive by adding, replacing or deleting any applicable request parameters defined in the *additions* or *deletions* attributes of the referenced <primitiveProfile> resource.

If *notificationContentType* is set to "Modified Attributes" or "All Attributes", the Originator shall also apply the referenced <primitiveProfile> resource to the representation of the subscribed-to resource or the child resource of the subscribed-to resource that is included in the content of the notification, if applicable. Before doing this the Originator shall make the following checks:

* Check that the primitive profile’s *resourceTypes* and *resourceIDs* attributes match the subscribed-to resource or the child resource of the subscribed-to resource that is included in the content of the notification.
* Check that the primitive profile’s *operations* attribute includes the Notify operation and the *releaseVersions* attribute matches the release version indicator of the notification.
* Check that the *applicability* attribute includes the value “NOTIFICATIONS\_FROM\_CSE”.

If all these checks are successful, the Originator shall apply the <primitiveProfile> resource to the resource attributes included in the content of the notification by adding, replacing or deleting resource attributes defined in the *additions* and *deletions* attribute of the <primitiveProfile> resource.

In addition to the procedures above, if the event was caused by a request and if the *enableEventNotificationOriginator* attribute in the subscription is present and set to *true* then the *creator* attribute of the notification shall contain the originator of the request that caused the event.

Step 2.2 Check the *notificationEventCat* attribute:

* If the *notificationEventCat* attribute is set, the Notify request primitive shall employ the ***Event Category*** parameter as given in the *notificationEventCat* attribute. Then continue with the step 2.3.
* If the *notificationEventCat* attribute is not configured, then continue with step 2.3.

Step 2.3 Check the *latestNotify* attribute:

* If the *latestNotify* attribute is set, the Originator shall assign ***Event Category*** parameter of value “latest” of the notifications generated pertaining to the subscription created.

Step 2.4 Check the batching notifications policy and the *rateLimit* attribute:

* See details in oneM2M TS-0001 [6], clause 10.2.10.7.
* If both the *batchNotify* and *primitiveProfileID* attributes of the <subscription> resource are configured, the Originator shall attempt to apply the referenced <primitiveProfile> resource to the individual notifications embedded within an aggregated notification using the same procedure as described in Step 2.1. In addition, the Originator shall also apply the primitive profile to the request parameters of the aggregated notification request primitive by adding, replacing or deleting any applicable request parameters defined in the *additions* or *deletions* attributes of the referenced <primitiveProfile> resource

NOTE: The use of some attributes such as *preSubscriptionNotify* is not supported in the present document.

Step 2.5 Check the *notificationURI* attribute:

* The Originator shall fetch the *notificationURI* attribute and set the value to the ***To*** parameter of the Notify request. When the *notificationURI* attribute contains more than one target, the Originator shall generate each Notify request per target.
* If the *notificationURI* attribute includes the notification serialization indication, in form of key-value pair, e.g. "ct=json", after the delimiter "?", the Originator shall serialize the notification for the notification target in that serialization type. The delimiter with the serialization indication shall be removed when the target is set to the ***To*** parameter of the Notify request. Then continue with step 3.0.

Step 3.0 The Originator shall check the notification and reachability schedules, but the notification schedules may be checked in different order:

* If the <subscription> resource associated with the modified resource includes a <notificationSchedule> child resource, the Originator shall check the time periods given in the scheduleElement attribute of the <notificationSchedule> child resource.
* Also, the Originator shall check the reachability schedule associated with the Receiver by exploring its <schedule> resource. If reachability schedules are not present in a Node then that Entity is considered to be always reachable.
* If notificationSchedule and reachability schedule indicate that message transmission is allowed, then proceed with step 5.0. Otherwise, proceed with step 4.0.
* In particular, if the *notificationEventCat* attribute is set to ‘immediate’ and the <notificationSchedule> resource does not allow transmission, then go to step 5.0 and send the corresponding Notify request primitive by temporarily ignoring the Originator’s notification schedule.

Step 4.0 Check the *pendingNotification* attribute:

* If the *pendingNotification* attribute is set, then the Originator shall cache pending Notify request primitives according to the *pendingNotification* attribute. The possible values are ‘sendLatest’ and ‘sendAllPending’. If the value of pendingNotification is set to ‘sendLatest’, the most recent Notify request primitive shall be cached by the Originator and it shall set the ***Event Category*** parameter to “latest”. If *pendingNotification* is set to ‘sendAllPending’, all Notify request primitives shall be cached by the Originator. If the *pendingNotification* attribute is not configured, the Originator shall discard the corresponding Notify request primitive. Any cached Notify request primitives are sent to the Receiver once message transmission becomes possible (see the step 6.0).

Step 5.0 Check the *expirationCounter* attribute:

* 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 <subscription> resource shall be deleted. Then end the ‘Compose Notify Request Primitive’ procedure.
* If the *expirationCounter* attribute is not configured, then end the ‘Compose Notify Request Primitive’ procedure.

When message transmission becomes possible, the Originator shall execute the following steps in order:

Step 6.0 If the *pendingNotification* attribute is set, the Originator shall send any cached Notify request primitives and then continue with the step 7.0

Step 7.0 Check the *expirationCounter* attribute:

* If the *expirationCounter* attribute is set, then its value shall be decreased by one when the Originator successfully sends the Notify request primitive. If the counter meets zero, the corresponding <subscription> resource shall be deleted. Then end the ‘Compose Notify Request Primitive’ procedure.
* If the *expirationCounter* attribute is not configured, then end the ‘Compose Notify Request Primitive’ procedure.

***Receiver:***

When the Hosting CSE receives a Notify request primitive, the Hosting CSE shall check validity of the primitive parameters. In case the Receiver is a transit CSE which forwards or aggregates Notify request primitives before sending to the subscriber or other transit CSEs, upon receiving the Notify request primitive with the ***Event Category*** parameter set to “latest”, the Receiver shall identify the latest Notify request primitive with the same subscription reference while storing Notify request primitives locally. When the Receiver as a transit CSE needs to send pending Notify request primitives, it shall send the latest Notify request primitive. When the Receiver as a transit CSE needs to send Notify request primitives, it shall use one of the serializations specified in the subscriber or other transit CSE *contentSerialization* attribute. If there is no *contentSerialization* value specified the transit CSE may use any serialization format.

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

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of Change 2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### 7.4.8.1 Introduction

The <subscription> resource contains subscription information for its subscribed-to resource. The subscription resource is a child of the subscribed to resource.

The detailed description can be found in clause 9.6.8 in oneM2M TS-0001 [6].

Table 7.4.8.1‑1: Data type definition of <subscription> resource

|  |  |  |
| --- | --- | --- |
| Data Type ID | File Name | Note |
| subscription | CDT-subscription.xsd |  |

Table 7.4.8.1‑2: Universal/Common Attributes of <subscription> 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 7.4.8.1‑3: Resource Specific Attributes of <subscription> resource

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Request Optionality  | Data Type | Default Value and Constraints |
| Create | Update |
| *eventNotificationCriteria* | O | O | m2m:eventNotificationCriteria  | Default behaviour is notification on Update\_of\_Resource |
| *expirationCounter* | O | O | xs:positiveInteger | No default |
| *notificationURI* | M | O | list of xs:anyURI | No defaultIn this value, it may contain notification serialization type (i.e. xml, json, cbor) per target. This shall be applied only for the URL formatted target (c.f. resource ID). When the type is set, only one type indication shall be appended in the target as the key-value format with delimiter "?",. If the value already contains "?" character for application queries, the type information shall be appended with "&". The key shall be "ct" (content serialization type). Note that this serialization type is in lower cases.Examples: http://mydomain/notificationHandler?ct=jsonhttp://mydomain/notificationHandler?q=true&ct=json |
| *groupID* | O | O | xs:anyURI | No default |
| *notificationForwardingURI* | O | O | xs:anyURI | No default |
| *batchNotify* | O | O | m2m:batchNotify | No default |
| *rateLimit* | O | O | m2m:rateLimit | No default |
| *preSubscriptionNotify* | O | NP | xs:positiveInteger | No default |
| *pendingNotification* | O | O | m2m:pendingNotification | No default |
| *notificationStoragePriority* | O | O | xs:positiveInteger | No default |
| *latestNotify* | O | O | xs:boolean | No default |
| *notificationContentType* | O | O | m2m:notificationContentType | See TS-0001 [6]. Table 9.6.8-4. |
| *notificationEventCat* | O | O | m2m:eventCat | No default |
| *subscriberURI* | O | NP | xs:anyURI | No default |
| *associatedCrossResourceSub* | O | O | m2m:listOfURIs  | No default. This attribute shall only be modified by a <crossResourceSubscription> Hosting CSE. described in clause 7.4.58 |
| *notificationStatsEnable* | O | O | xs:boolean | false |
| *notificationStatsInfo* | NP | NP | m2m:setOfNotificationStatsInfo | No default |
| *primitiveProfileID* | O | O | xs:anyURI | No default |
| *enableEventNotificationOriginator* | O | O | xs:boolean | false |

Table 7.4.8.1‑4: Reference of child resources

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

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of Change 2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of Change 2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 8.2.3 Resource attributes

In protocol bindings, resource attributes names shall be translated into short names shown in the following tables.

Table 8.2.3‑1: Resource attribute short names (1/6)

| Attribute Name | Occurs in | Short Name |
| --- | --- | --- |
| *accessControlPolicyIDs* | All except accessControlPolicy, contentInstance | ***acpi*** |
| *announcedAttribute* | All announceable resources | ***aa*** |
| *announceTo* | All announceable resources | ***at*** |
| *announceSyncType* | All announceable resources | ***ast*** |
| *creationTime* | All | ***ct*** |
| *expirationTime* | All except contentInstance, CSEBase | ***et*** |
| labels | All (optional) | ***lbl*** |
| *lastModifiedTime* | All | ***lt*** |
| *link* | All | ***lnk*** |
| *parentID* | All | ***pi*** |
| *resourceID* | All | ***ri*** |
| resourceType | All | ***ty\**** |
| *stateTag* | container, contentInstance, delivery, request, flexContainer | ***st*** |
| *resourceName* | All | ***rn*** |
| *privileges* | accessControlPolicy | ***pv*** |
| *selfPrivileges* | accessControlPolicy | ***pvs*** |
| *authorizationDecisionResourceIDs* | accessControlPolicy | ***adri*** |
| *authorizationPolicyResourceIDs* | accessControlPolicy | ***apri*** |
| *authorizationInformationResourceIDs* | accessControlPolicy | ***airi*** |
| *App-ID* | AE | ***api*** |
| *AE-ID* | AE | ***aei*** |
| *AE-IDList* | AEContactListPerCSE | ***ail*** |
| *appName* | AE | ***apn*** |
| *pointOfAccess* | AE, CSEBase, remoteCSE | ***poa*** |
| *ontologyRef* | AE, container, contentInstance, semanticDescriptor. flexContainer, timeSeries, reasoningRules | ***or*** |
| *nodeLink* | AE, CSEBase, remoteCSE, flexContainer | ***nl*** |
| contentSerialization | AE | ***csz*** |
| *registrationStatus* | AE | ***regs*** |
| *trackRegistrationPoints* | AE | ***trps*** |
| *sessionCapabilities* | AE | ***scp*** |
| *activityPatternElements* | AE, remoteCSE | ***ape*** |
| *triggerEnable* | AE, remoteCSE | ***tren*** |
| *enableTimeCompensation* | AE, remoteCSE | ***etc*** |
| *creator* | container, contentInstance, eventConfig, group, pollingChannel, statsCollect, statsConfig, subscription, semanticDescriptor, notificationTargetPolicy, flexContainer, timeSeries, crossResourceSubscription, backgroundDataTransfer | ***cr*** |
| *maxNrOfInstances* | container, timeSeries, flexContainer | ***mni*** |
| *maxByteSize* | container, timeSeries, flexContainer | ***mbs*** |
| *maxInstanceAge* | container, timeSeries, flexContainer | ***mia*** |
| *currentNrOfInstances* | container, timeSeries, flexContainer | ***cni*** |
| *location* | *CSEBase*, *remoteCSE*, *AE*, *container*, *flexContainer*, *timeSeries, deviceInfo* | ***loc*** |
| *custodian* | All except contentInstance, timeSeriesInstance, flexContainerInstance, accessControlPolicy, pollingChannel | ***cstn*** |

Table 8.2.3‑2: Resource attribute short names (2/6)

| Attribute Name | Occurs in | Short Name |
| --- | --- | --- |
| *currentByteSize* | container,timeSeries, flexContainer | ***cbs*** |
| *locationID* | container | ***li*** |
| *disableRetrieval* | container | ***disr*** |
| *contentInfo* | contentInstance, timeSeries | ***cnf*** |
| *contentSize* | contentInstance, timeSeriesInstance, flexContainer, flexContainerInstance | ***cs*** |
| *contentRef* | contentInstance | ***conr*** |
| *containerDefinition* | flexContainer | ***cnd*** |
| *primitiveContent*  | request | ***pc\**** |
| *content* | contentInstance, timeSeriesInstance | ***con*** |
| *deletionCnt* | contentInstance | ***dcnt*** |
| *cseType* | CSEBase, remoteCSE | ***cst*** |
| *CSE-ID* | CSEBase, remoteCSE, service SubscribedNode, AEContactListPerCSE | ***csi*** |
| *supportedResourceType* | CSEBase | ***srt*** |
| *notificationCongestionPolicy* | CSEBase | ***ncp*** |
| *currentTime* | CSEBase | ***ctm*** |
| *source* | delivery | ***sr*** |
| *target* | delivery, request | ***tg*** |
| *lifespan* | delivery | ***ls*** |
| *eventCat* | delivery | ***ec*** |
| *deliveryMetaData* | delivery | ***dmd*** |
| *aggregatedRequest* | delivery | ***arq*** |
| *eventID* | eventConfig, statsCollect | ***evi*** |
| *eventType* | eventConfig | ***evt*** |
| *evenStart* | eventConfig | ***evs*** |
| *eventEnd* | eventConfig | ***eve*** |
| *operationType* | eventConfig | ***opt*** |
| *dataSize* | eventConfig | ***ds*** |
| *eventResourceTypes* | eventConfig | ***erts*** |
| *eventResourceIDs* | eventConfig | ***eris*** |
| *execStatus* | execInstance | ***exs*** |
| *execResult* | execInstance | ***exr*** |
| *execDisable* | execInstance | ***exd*** |
| *execTarget* | execInstance, mgmtCmd | ***ext*** |
| *execMode* | execInstance, mgmtCmd | ***exm*** |
| *execFrequency* | execInstance, mgmtCmd | ***exf*** |
| *execDelay* | execInstance, mgmtCmd | ***exy*** |
| *execNumber* | execInstance, mgmtCmd | ***exn*** |
| *execReqArgs* | execInstance, mgmtCmd | ***exra*** |
| *execEnable* | mgmtCmd | ***exe*** |
| *memberType* | group | ***mt*** |
| *specializationType* | group | ***spty*** |
| *currentNrOfMembers* | group | ***cnm*** |
| *maxNrOfMembers* | group | ***mnm*** |
| *memberIDs* | group, backgroundDataTransfer | ***mid*** |
| *membersAccessControlPolicyIDs* | group | ***macp*** |
| *memberTypeValidated* | group | ***mtv*** |
| *consistencyStrategy* | group | ***csy*** |
| *semanticSupportIndicator* | group | ***ssi*** |
| *notifyAggregation* | group | ***nar*** |
| *groupName* | group, subscription | ***gn*** |
| *locationSource* | locationPolicy | ***los*** |
| *locationUpdatePeriod* | locationPolicy | ***lou*** |
| *locationTargetID* | locationPolicy | ***lot*** |
| *locationServer* | locationPolicy | ***lor*** |
| *locationContainerID* | locationPolicy | ***loi*** |
| *locationContainerName* | locationPolicy | ***lon*** |
| *locationStatus* | locationPolicy | ***lost*** |
| *authID* | locationPolicy | ***aid*** |
| *retrieveLastKnownLocation* | locationPolicy | ***rlkl*** |
| *locationUpdateEventCriteria* | locationPolicy | ***luec*** |
| *locationInformationType* | locationPolicy | ***lit*** |
| *geographicalTargetArea* | locationPolicy | ***gta*** |
| *geofenceEventCriteria* | locationPolicy | ***gec*** |
| *description* | mgmtCmd, mgmtObj, all management resources from firmware, ontology, ontologyMappingAlgorithm | ***dc*** |
| *cmdType* | mgmtCmd | ***cmt*** |
| *mgmtDefinition* | mgmtObj, all management resources from firmware | ***mgd*** |
| *objectIDs* | mgmtObj | ***obis*** |

Table 8.2.3‑3: Resource attribute short names (3/6)

| Attribute Name | Occurs in | Short Name |
| --- | --- | --- |
| *objectPaths* | mgmtObj | ***obps*** |
| *mgmtSchema* | mgmtObj | ***mgs*** |
| *nodeID* | node | ***ni*** |
| *hostedCSELink* | node | ***hcl*** |
| *mgmtClientAddress* | node | ***mgca*** |
| *hostedAELinks* | node | ***hael*** |
| *hostedServiceLinks* | node | ***hsl*** |
| *networkID* | node | ***nid*** |
| *roamingStatus* | node | ***rms*** |
| *nodeType* | node | ***nty*** |
| *requestAggregation* | pollingChannel | *rqag* |
| *CSEBase* | remoteCSE | ***cb\**** |
| *M2M-Ext-ID* | remoteCSE, AE, locationPolicy, triggerRequest | ***mei*** |
| *M2M-Ext-IDs* | nwMonitoringReq | ***meis*** |
| *Trigger-Recipient-ID* | remoteCSE, triggerRequest, AE | ***tri*** |
| *requestReachability* | remoteCSE | ***rr*** |
| *triggerReferenceNumber* | remoteCSE, AE | ***trn*** |
| *descendantCSEs* | remoteCSE | ***dcse*** |
| *multicastCapability* | remoteCSE | ***mtcc*** |
| *originator* | request, flexContainerInstance | ***org*** |
| *metaInformation* | request | ***mi*** |
| *requestStatus* | request | ***rs*** |
| *operationResult* | request | ***ors*** |
| *operation* | request | ***op\**** |
| *requestID* | request | ***rid*** |
| *scheduleElement* | schedule | ***se*** |
| *networkCoordinated* | schedule | ***nco*** |
| *deviceIdentifier* | serviceSubscribedNode | ***di*** |
| *ruleLinks* | serviceSubscribedNode | ***rlk*** |
| *niddRequired* | serviceSubscribedNode | ***nrq*** |
| *statsCollectID* | statsCollect | ***sci*** |
| *collectingEntityID* | statsCollect | ***cei*** |
| *collectedEntityID* | statsCollect | ***cdi*** |
| *devStatus* | areaNwkDeviceInfo | ***ss*** |
| *statsRuleStatus* | statsCollect | ***srs*** |
| *statModel* | statsCollect | ***sm*** |
| *collectPeriod* | statsCollect | ***cp*** |
| *eventNotificationCriteria* | subscription | ***enc*** |
| *expirationCounter* | subscription, crossResourceSubscription | ***exc*** |
| *notificationURI* | subscription, crossResourceSubscription | ***nu*** |
| *groupID* | subscription | ***gpi*** |
| *notificationForwardingURI* | subscription | ***nfu*** |
| *batchNotify* | subscription | ***bn*** |
| *rateLimit* | subscription | ***rl*** |
| *preSubscriptionNotify* | subscription | ***psn*** |
| *pendingNotification* | subscription | ***pn*** |
| *notificationStoragePriority* | subscription | ***nsp*** |
| *latestNotify* | subscription | ***ln*** |
| *notificationContentType* | subscription | ***nct*** |
| *notificationEventCat* | subscription, crossResourceSubscription | ***nec*** |
| *subscriberURI* | subscription, crossResourceSubscription | ***su*** |
| *enableEventNotificationOriginator* | subscription | ***eeno*** |
| *version* | firmware, software, token, softwareCampaign | ***vr*** |
| *URL* | firmware, software, softwareCampaign | ***url*** |
| *update* | firmware | ***ud*** |
| *updateStatus* | firmware | ***uds*** |
| *install* | software | ***in*** |
| *uninstall* | software | ***un*** |
| *installStatus* | software | ***ins*** |
| *activate* | software, m2mServiceSubscriptionProfile | ***act*** |
| *deactivate* | software | ***dea*** |
| *activeStatus* | software, areaNwkInfo | ***acts*** |
| *memAvailable* | memory | ***mma*** |
| *memTotal* | memory | ***mmt*** |

Table 8.2.3‑4: Resource attribute short names (4/6)

| Attribute Name | Occurs in | Short Name |
| --- | --- | --- |
| *areaNwkType* | areaNwkInfo | ***ant*** |
| *listOfDevices* | areaNwkInfo | ***ldv*** |
| *devId* | areaNwkDeviceInfo | ***dvd*** |
| *devType* | areaNwkDeviceInfo | ***dvt*** |
| *areaNwkId* | areaNwkDeviceInfo | ***awi*** |
| *sleepInterval* | areaNwkDeviceInfo | ***sli*** |
| *sleepDuration* | areaNwkDeviceInfo | ***sld*** |
| *listOfNeighbors* | areaNwkDeviceInfo | ***lnh*** |
| *batteryLevel* | battery | ***btl*** |
| *batteryStatus* | battery | ***bts*** |
| *deviceLabel* | deviceInfo | ***dlb*** |
| *manufacturer* | deviceInfo | ***man*** |
| *model* | deviceInfo | ***mod*** |
| *deviceType* | deviceInfo | ***dty*** |
| *fwVersion* | deviceInfo | ***fwv*** |
| *swVersion* | deviceInfo | ***swv*** |
| *hwVersion* | deviceInfo | ***hwv*** |
| *manufacturerDetailsLink* | deviceInfo | ***mfdl*** |
| *manufacturingDate* | deviceInfo | ***mfd*** |
| *subModel* | deviceInfo | ***smod*** |
| *deviceName* | deviceInfo | ***dvnm*** |
| *osVersion* | deviceInfo | ***osv*** |
| *country* | deviceInfo | ***cnty*** |
| *systemTime* | deviceInfo | ***syst*** |
| *supportURL* | deviceInfo | ***spur*** |
| *presentationURL* | deviceInfo | ***purl*** |
| *protocol* | deviceInfo | ***ptl*** |
| *capabilityName* | deviceCapability | ***can*** |
| *attached* | deviceCapability | ***att*** |
| *capabilityActionStatus* | deviceCapability | ***cas*** |
| *enable* | deviceCapability, allJoynSvcObject | ***ena*** |
| *disable* | deviceCapability | ***dis*** |
| *currentState* | deviceCapability | ***cus*** |
| *reboot* | reboot | ***rbo*** |
| *factoryReset* | reboot | ***far*** |
| *logTypeId* | eventLog | ***lgt*** |
| *logData* | eventLog | ***lgd*** |
| *logStatus* | eventLog | ***lgst*** |
| *logStart* | eventLog | ***lga*** |
| *logStop* | eventLog | ***lgo*** |
| *firmwareName* | firmware | ***fwn*** |
| *softwareName* | software, softwareCampaign | ***swn*** |
| *cmdhPolicyName* | cmdhPolicy | ***cpn*** |
| *mgmtLink* | cmdhPolicy, activeCmdhPolicy, cmdhDefaults, cmdhNetworkAccessRules, cmdhNwAccessRule | ***cmlk*** |
| *activeCmdhPolicyLink* | activeCmdhPolicy | ***acmlk*** |
| *order* | cmdhDefEcValue, cmdhLimits | ***od*** |
| *defEcValue* | cmdhDefEcValue | ***dev*** |
| *requestOrigin* | cmdhDefEcValue, cmdhLimits | ***ror*** |
| *requestContext* | cmdhDefEcValue, cmdhLimits | ***rct*** |
| *requestContextNotification* | cmdhDefEcValue, cmdhLimits | ***rctn*** |
| *requestCharacteristics* | cmdhDefEcValue, cmdhLimits | ***rch*** |
| *applicableEventCategories* | cmdhNetworkAccessRules | ***aecs*** |
| *applicableEventCategory* | cmdhEcDefParamValues, cmdhBuffer | ***aec*** |
| *defaultRequestExpTime* | cmdhEcDefParamValues | ***dqet*** |
| *defaultResultExpTime* | cmdhEcDefParamValues | ***dset*** |
| *defaultOpExecTime* | cmdhEcDefParamValues | ***doet*** |
| *defaultRespPersistence* | cmdhEcDefParamValues | ***drp*** |
| *defaultDelAggregation* | cmdhEcDefParamValues | ***dda*** |
| *limitsEventCategory* | cmdhLimits | ***lec*** |
| *limitsRequestExpTime* | cmdhLimits | ***lqet*** |
| *limitsResultExpTime* | cmdhLimits | ***lset*** |
| *limitsOpExecTime* | cmdhLimits | ***loet*** |
| *limitsRespPersistence* | cmdhLimits | ***lrp*** |
| *limitsDelAggregation* | cmdhLimits | ***lda*** |
| *targetNetwork* | cmdhNwAccessRule | ***ttn*** |

Table 8.2.3‑5: Resource attribute short names (5/6)

| Attribute Name | Occurs in | Short Name |
| --- | --- | --- |
| *minReqVolume* | cmdhNwAccessRule | ***mrv*** |
| *spreadingWaitTime* | cmdhNwAccessRule | ***swt*** |
| *backOffParameters* | cmdhNwAccessRule | ***bop*** |
| *otherConditions* | cmdhNwAccessRule | ***ohc*** |
| *maxBufferSize* | cmdhBuffer | ***mbfs*** |
| *storagePriority* | cmdhBuffer | ***sgp*** |
| *applicableCredIDs* | serviceSubscribedAppRule | ***apci*** |
| *allowedApp-IDs* | serviceSubscribedAppRule | ***aai*** |
| *allowedAEs* | serviceSubscribedAppRule | ***aae*** |
| *allowedRole-IDs* | serviceSubscribedAppRule | ***ari*** |
| *notificationTargetURI* | notificationTargetMgmtPolicyRef | ***ntu*** |
| *notificationPolicyID* | notificationTargetMgmtPolicyRef | ***npi*** |
| *action* | notificationTargetPolicy | ***acn*** |
| *policyLabel* | notificationTargetPolicy | ***plbl*** |
| *rulesRelationship* | notificationTargetPolicy | ***rrs*** |
| *creator* | notificationTargetPolicy | ***cr*** |
| *deletionRules* | policyDeletionRules | ***dr*** |
| *deletionRulesRelation* | policyDeletionRules | ***drr*** |
| *dynamicAuthorizationConsultationIDs* | All resources having an accessControlPolicyID attribute | ***daci*** |
| *dynamicAuthorizationEnabled* | dynamicAuthorizationConsultation | ***dae*** |
| *dynamicAuthorizationPoA* | dynamicAuthorizationConsultation | ***dap*** |
| *dynamicAuthorizationLifetime* | dynamicAuthorizationConsultation | ***dal*** |
| *descriptorRepresentation* | semanticDescriptor | ***dcrp*** |
| *semanticOpExec* | semanticDescriptor | ***soe*** |
| *descriptor* | semanticDescriptor | ***dsp*** |
| *relatedSemantics* | semanticDescriptor | ***rels*** |
| *semanticValidated* | semanticDescriptor | ***svd*** |
| *validationEnable* | semanticDescriptor | ***vlde*** |
| *periodicInterval* | timeSeries | ***pei*** |
| *missingDataDetect* | timeSeries | ***mdd*** |
| *missingDataMaxNr* | timeSeries | ***mdn*** |
| *missingDataList* | timeSeries | ***mdlt*** |
| *missingDataCurrentNr* | timeSeries | ***mdc*** |
| *missingDataDetectTimer* | timeSeries | ***mdt*** |
| *dataGenerationTime* | timeSeriesInstance | ***dgt*** |
| *sequenceNr* | timeSeriesInstance | ***snr*** |
| *roleID* | role | ***rlid*** |
| *roleName* | role | ***rlnm*** |
| *tokenLink* | role | ***rltl*** |
| *tokenID* | token | ***tkid*** |
| *tokenObject* | token | ***tkob*** |
| *issuer* | token, role | ***tkis*** |
| *holder* | token, role | ***tkhd*** |
| *notBefore* | token, role | ***tknb*** |
| *notAfter* | token, role | ***tkna*** |
| *tokenName* | token | ***tknm*** |
| *audience* | token | ***tkau*** |
| *permissions* | token | ***tkps*** |
| *extension* | token | ***tkex*** |
| *e2eSecInfo* | CSEBase, remoteCSE, AE | ***esi*** |
| *supportedReleaseVersions* | CSEBase, remoteCSE, AE | ***srv*** |

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, AE | ***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 3 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*