|  |
| --- |
|  |

|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* | SDS #39 |
| Source:\* | Bob Flynn, Convida Wireless, [Flynn.Bob@ConvidaWireless.com](mailto:Flynn.Bob@ConvidaWireless.com) |
| Date:\* | 2019-02-18 |
| Reason for Change/s:\* | Subscription with Blocking Notification Event Type changes |
| CR against: Release\* | Release 3 |
| CR against: WI\* | Active <Work Item number>  MNT maintenance / < Work Item number(optional)>  Is this a mirror CR? Yes  No  mirror CR number: (Note to Rapporteur - use latest agreed revision)  STE Small Technical Enhancements / < Work Item number (optional)>  Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS0004 v3\_9\_3 |
| Clauses \* | 6.6.3.5,7.4.8.2,7.5.1.2,7.5.1.2.1,7.5.1.2.20,7.5.2, |
| Type of change: \* | Editorial change  Bug Fix or Correction  Change to existing feature or functionality  New feature or functionality  Only ONE of the above shall be ticked |
| Impacted other TS/TR(s) |  |
| Post Freeze checking:\* | This CR contains only essential changes and corrections? YES  NO  This CR may break backwards compatibility with the last approved version of the TS? YES  NO |
| Template Version: January 2017 (Do not modify) | |

**oneM2M Notice**

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

GUIDELINES for Change Requests:

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

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

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

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

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

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

Follow the drafting rules.

All pictures must be editable.

Check spelling and grammar to the extent practicable.

Use Change bars for modifications.

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

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

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

## Introduction

R03 – remove concept of blocking on an attribute and block on the resource.

R02 – Restructure notify in change 4.

This is a PRO contribution for

|  |  |
| --- | --- |
| ARC-2017-0304R03-Block\_update\_during\_notification | Qualcomm |
| ARC-2018-0175R01-Subscription\_NotificationEventType\_Blocking\_R3 | Convida |

For <subscription> resource, TS-0001 Table 9.6.8-3 describes *notificationEventType* attribute setting of type G – “Update to attributes of the subscribed-to resource with blocking of the triggering UPDATE operation”

This contribution is for the corresponding changes to TS-0004.

Change 1 – Adds new error codes BLOCKING\_SUBSCRIPTION\_ALREADY\_EXISTS.

Change 2 – Adds procedure to Create/Update <subscription> to check that this type of subscription does not exist for the same resource/attribute.

-aligns procedures with 7.2.2.2

-specifies a specific error code for subscription verification

Change 3 – add bullet item to list of notification types

Change 4 – Adds new notification procedure (generating a blocking notification)

Change 5 – Add a “response primitive” to the list of valid values in a “response primitive”

Change 6 – add enumerations to m2m:notificationEventType

Change 7 – add changes to general UPDATE resource procedures

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

#### Originator error response class

Table 6.6.3.5-1 specifies the RSCs for Originator error responses.

41xx codes are oneM2M specific.

Table 6.6.3.5‑1: RSCs for Originator error response class

|  |  |
| --- | --- |
| Numeric Code | Description |
| 4000 | BAD\_REQUEST |
| 4001 | RELEASE\_VERSION\_NOT\_SUPPORTED |
| 4004 | NOT\_FOUND |
| 4005 | OPERATION\_NOT\_ALLOWED |
| 4008 | REQUEST\_TIMEOUT |
| 4015 | UNSUPPORTED\_MEDIA\_TYPE |
| 4101 | SUBSCRIPTION\_CREATOR\_HAS\_NO\_PRIVILEGE |
| 4102 | CONTENTS\_UNACCEPTABLE |
| 4103 | ORIGINATOR\_HAS\_NO\_PRIVILEGE |
| 4104 | GROUP\_REQUEST\_IDENTIFIER\_EXISTS |
| 4105 | CONFLICT |
| 4106 | ORIGINATOR\_HAS\_NOT\_REGISTERED |
| 4107 | SECURITY\_ASSOCIATION\_REQUIRED |
| 4108 | INVALID\_CHILD\_RESOURCE\_TYPE |
| 4109 | NO\_MEMBERS |
| 4110 | GROUP\_MEMBER\_TYPE\_INCONSISTENT |
| 4111 | ESPRIM\_UNSUPPORTED\_OPTION |
| 4112 | ESPRIM\_UNKNOWN\_KEY\_ID |
| 4113 | ESPRIM\_UNKNOWN\_ORIG\_RAND\_ID |
| 4114 | ESPRIM\_UNKNOWN\_RECV\_RAND\_ID |
| 4115 | ESPRIM\_BAD\_MAC |
| 4116 | ESPRIM\_IMPERSONATION\_ERROR |
| 4117 | ORIGINATOR\_HAS\_ALREADY\_REGISTERED |
| 4118 | ONTOLOGY\_NOT\_AVAILABLE |
| 4119 | LINKED\_SEMANTICS\_NOT\_AVAILABLE |
| 4120 | INVALID\_SEMANTICS |
| 4121 | MASHUP\_MEMBER\_NOT\_FOUND |
| 4122 | INVALID\_TRIGGER\_PURPOSE |
| 4123 | ILLEGAL\_TRANSACTION\_STATE\_TRANSITION\_ATTEMPTED |
| 4124 | BLOCKING\_SUBSCRIPTION\_ALREADY\_EXISTS |

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

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

#### 7.4.8.2 <subscription> resource specific procedure on CRUD operations

##### Create

***Originator****:*

No change from the generic procedures in clause 7.2.2.1.

***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.

1. Recv-6.2: The following steps are in addition to the procedures defined in clause 7.3.3.1.
   1. Check if the subscribed-to resource, addressed in ***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 "TARGET\_NOT\_SUBSCRIBABLE" error.
2. Recv-6.3: The following steps are in addition to the procedures defined in clause 7.3.3.15
   1. Check if the Originator has privileges for retrieving the subscribed-to resource. If the Originator does not have the privilege, the Hosting CSE shall return the response primitive with Response Status Code indicating "ORIGINATOR\_HAS\_NO\_PRIVILEGE" error.
3. Recv-6.4: The following steps are in addition to the procedures defined in clause 7.3.3.3
4. 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 ***Response Status Code*** indicating "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 ***Response Status Code*** indicating "BAD\_REQUEST" error.
   * If any other attributes of the <subscription> resource are specified the Hosting CSE shall return the primitive with ***Response Status Code*** indicating "BAD\_REQUEST" error.
   * If any condition tag of the *eventNotificationCriteria* attribute other than *attribute* condition tag is specified, the Hosting CSE shall return the response primitive with ***Response Status Code*** indicating "BAD\_REQUEST" error.
5. 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.
   * If the Hosting CSE cannot send the Subscription Verification request primitive, the Hosting CSE shall return the Create <subscription> response primitive with a ***Response Status Code*** indicating "SUBSCRIPTION\_VERIFICATION\_INITIATION\_FAILED" error.
   * If the Hosting CSE sent a Subscription Verification primitive, the Hosting CSE shall check if the Notify response primitive contains a ***Response Status Code*** indicating "OK". If not, the Hosting CSE shall return the Create <subscription> response primitive with a ***Response Status Code*** indicating “NOT\_ACCEPTABLE”.
6. If the *associatedCrossResourceSub* is provided, check that the Hosting CSE ID value in the *associatedCrossResourceSub* is the same as the ***From*** parameter of the request. If not, return the response primitive with a ***Response Status Code*** indicating “BAD\_REQUEST”.
7. Recv-6.5: The following steps are in addition to the procedures defined in clause 7.3.3.5
8. If the Originator does not provide notificationContentType, the Hosting CSE shall set it as ‘all attributes'.
9. If the *notificationURI* is not the Originator, the Hosting CSE shall store the Originator ID as the <subscription> resource's *creator* attribute.
10. 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.

##### Retrieve

***Originator****:*

No change from the generic procedures in clause 7.2.2.1.

***Receiver****:*

No change from the generic procedures in clause 7.2.2.2.

##### Update

***Originator****:*

The following change from the generic procedures in clause 7.2.2.1.

1. Orig-1.0: The originator shall not specify *notficationEventType* set to “Blocking\_Update”.

***Receiver****:*

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

1. Recv-6.4: The following steps are in addition to the procedures defined in clause 7.3.3.4
2. Check if the *notificationEventType* is set to ‘Blocking\_Update‘. If so, the Hosting CSE shall return the response primitive with ***Response Status Code*** indicating “BAD\_REQUEST" error.
3. Recv-6.5. Check if a new *associatedCrossResourceSub* is provided.
   1. If so, check that the Hosting CSE ID value in the *associatedCrossResourceSub* is the same as the ***From*** parameter of the request. If a <crossResourceSubscription> Hosting CSE ID is removed from *associatedCrossResourceSub*, the Hosting CSE shall send a Notify request for Subscription Deletion using the procedures in 7.5.1.2.4. to the <crossResourceSubscription> Hosting CSE*.*

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

### -----------------------Start of change 3-------------------------------------------

#### Notification procedures

##### 7.5.1.2.1 Introduction

Notification procedures shall be employed for the following use cases:

* to notify Receiver(s) of modifications of a resource for an associated <subscription> resource;
* to request Receiver(s) to perform resource subscription verification;
* to notify deletion of the <subscription> resource;
* to notify Receiver(s) for Asynchronous Non-blocking Request;
* to notify Receiver(s) of modifications of a resource when the subscription relationship is established through the <group> resource;
* to send the response corresponding to a request delivered via service layer long polling (clause 7.4.22.2.2 Retrieve <pollingChannelURI>);
* to notify Receiver(s)(i.e. IPE) for on-demand discovery request;
* to notify Receiver(s) of the missing Time Series Data points for an associated <subscription> resource;
* to notify Receiver(s) of a security related request (e.g. dynamic authorization and end-to-end security);
* to notify Receivers that an AE has changed registration point;
* to notify an IN-CSE that the Originator has a new/updated/deleted reference to an Application Entity Resource identifier;
* to notify Receiver(s) of a cross-resource notification generated by a <crossResourceSubscription> Hosting CSE.
* to notify Receiver that a trigerred update on the subscribed-to resource has been blocked and retargeted to the receiver.

The following clauses specify the notification procedures for each of the above use cases.

### -----------------------End of change 3---------------------------------------------

### -----------------------Start of change 4-------------------------------------------

##### 7.5.1.2.19 Notification for Subscription Blocking Triggered update

Whenever the Hosting CSE receives an update request primitive for a target resource which has subscription with *notificationEventType* set to “Blocking\_Update”, it shall perform the below steps before Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation” is performed.

1. If the *attribute* condition tag of the *eventNotificationCriteria* attribute of the <subscription> resource is set, check that the *attribute* condition tag matches the modified attributes in the received UDATE request.
2. Prevent or block all other UPDATE request primitives to this target resource.
3. Create a Notification Request primitive and configure the request parameters as follows.
   1. Set the representation attribute of the notification to the representation of the target resource contained in the received UPDATE request primitive
4. Send Notification request primitive to the target specified in *notificationURI.*
5. Wait for a Notification response.
6. Process the Notification Response primitive
   1. If the notification ***Response Status Code*** is not successful,

return a response to the original blocked UPDATE request primitive with a ***Response Status Code*** indicating NOT\_ACCEPTABLE.

* 1. If the notification ***Response Status Code*** is successful, perform Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation” for the received UPDATE request.

1. Allow all other UPDATE request primitives for this target resource.

### -----------------------End of change 4---------------------------------------------

### -----------------------Start of change 5---------------------------------------------

### 7.5.2 Elements contained in the Content primitive parameter

Clauses 7.2.1.1 and 7.2.1.2 enumerate the forms that the ***Content*** primitive parameter takes in various Request and Response cases. Note that the ***Content*** primitive parameter is denoted as primitiveContent in both, CDT-requestPrimitive-v3\_9\_0.xsd and CDT-responsePrimitive-v3\_9\_0.xsd.

This clause details the Objects (elements) used in some of these cases. in the tables below.

The following elements are defined for use in the ***Content*** parameter of a request:

Table 7.5.2‑1: Elements used for request content

|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Applicable Operations** | **Data Type** | **Defined in** |
| m2m:<resourceType>  {other namespace identifier}:<resourceType> | C U | See element declaration | CDT-<resourceType>-v3\_9\_0.xsd |
| m2m:notification | N | m2m:notification | CDT-notification-v3\_9\_0.xsd |
| m2m:aggregatedNotification | N | m2m:aggregatedNotification | CDT-notification-v3\_9\_0.xsd |
| m2m:securityInfo | N | m2m:securityInfo | CDT-notification-v3\_9\_0.xsd |
| m2m:attributeList | R | m2m:attributeList | CDT-requestPrimitive-v3\_9\_0.xsd |
| m2m:responsePrimitive | N | Anonymous data type defined in the responsePrimitive declaration | CDT-responsePrimitive-v3\_9\_0.xsd |

The following elements are defined for use in the ***Content*** parameter of a response sent in reply to a request message with ***Operation*** and ***Result Content*** (rcn) parameters as given in the column "Applicable Operations" (the settings of the ***Result Content*** parameters are defined in clause 6.3.4.2.7; NP means the rcn parameter is not present).

Table 7.5.2‑2: Elements used for response content

|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Applicable Operations/rcn** | **Data Type** | **Element is Defined in** |
| m2m:<resourceType>  {other namespace identifier}:<resourceType>  See note 6 | C/1,9,NP  R/1,4,5,6,7,8,NP  U/1,9,NP  D/1,4,5,6,8  See note 1 | See element declaration | CDT-<resourceType>-v3\_9\_0.xsd |
| m2m:resource | C/3 | m2m:resourceWrapper | CDT-responsePrimitive-v3\_9\_0.xsd |
| m2m:URIList | R/NP  See note 2 | list of xs:anyURI | CDT-responsePrimitive-v3\_9\_0.xsd |
| m2m:resourceRefList | R/6  See note 2 | m2m:listOfChildResourceRef | CDT-responsePrimitive-v3\_9\_0.xsd |
| m2m:aggregatedResponse | C R U D  See note 3 | m2m:aggregatedResponse | CDT-responsePrimitive-v3\_9\_0.xsd |
| m2m:URI | C/2  See note 4 | xs:anyURI | CDT-responsePrimitive-v3\_9\_0.xsd |
| m2m:requestPrimitive | See note 7 | Anonymous data type defined in the requestPrimitive declaration | CDT-requestPrimitive-v3\_9\_0.xsd |
| m2m:debugInfo | See note 5 | xs:string | CDT-responsePrimitive-v3\_9\_0.xsd |
| m2m:securityInfo | N/NP | m2m:securityInfo | CDT-notification-v3\_9\_0.xsd |
| m2m:queryResult | R/10  See note 8 | xs:string | CDT-responsePrimitive-v3\_9\_0.xsd |
| NOTE 1: The case rcn = 7 applies to Retrieve operation only (R/7). It retrieves the original resource in case the To parameter points to an announced resource. The case R/NP applies to Retrieve operation (Non-Discovery) only.  NOTE 2: This applies to discovery operation only. For discovery, the format of the address (structured, unstructured) depends on the ***Discovery Result Type*** parameter setting (see clause 6.3.4.2.8).  NOTE 3: This applies to CRUD operations on a <fanOutPoint> child resource of a <group> parent resource. The ***Content*** parameter of each response primitive included in aggregatedResponse is set as given in one of the other rows of this table.  NOTE 4: This also applies to the response ("acknowledgement") to non-blocking requests in asynchronous and synchronous modes for any CRUD operation.  NOTE 5: This is a plain text messages which can optionally be included as debugging information in error responses. The language and content of the message is determined by the Service Provider.  NOTE 6: "{other namespace identifier}" refers to a namespace other than m2m.  NOTE 7: This applies to a polling response that contains a request for polling mechanism (see clause 7.4.22.2.2).  NOTE 8: This applies to semantic query operation only. The Originator may use the Accept option to indicate which media types are acceptable for the semantic query result, e.g. application/sparql-results+xml, or application/sparql-results+json. | | | |

The XML schema definition of the ***Content*** primitive parameter (i.e. datatype m2m:primitiveContent) allows to include XML wildcard elements. An XML representation of the ***Content*** primitive parameter shall include a root element which is associated with an XSD Global Element. The root element shall be prefixed with a namespace prefix identifier (e.g. *m2m:*) specified in the associated XSD which defines the respective Global Element. The ***Content*** primitive parameter allows to include namespaces other than m2m.

### -----------------------End of change 5---------------------------------------------

### -----------------------Start of change 6---------------------------------------------



##### m2m:notificationEventType

Used for ***eventNotificationCriteria*** conditions and ***notificationEvent*** element.

Table 6.3.4.2.19‑1: Interpretation of notificationEventType

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | Update\_of\_Resource | Default |
| 2 | Delete\_of\_Resource |  |
| 3 | Create\_of\_Direct\_Child\_Resource |  |
| 4 | Delete\_of\_Direct\_Child\_Resource |  |
| 5 | Retrieve\_of\_Container\_Resource\_With\_No\_Child\_Resource | Context: A RETRIEVE request targets a subscribed-to <container> resource with the Result Content parameter set to either "child-resources" or "attributes+child-resources".  A notification is initiated if the <contentInstance> child resource is obsolete or not present in the targeted parent resource. |
| 6 | Trigger\_Received\_For\_AE\_Resource | Context: A notification is initiated when a Trigger is Received by a Registrar CSE targeting the AE-ID associated with the <*AE*> resource of a Registree AE. |
| 7 | Blocking\_Update |  |

### -----------------------End of change 6---------------------------------------------

### -----------------------Start of change 7---------------------------------------------

#### Update the resource

If the Operation Execution Time parameter is given in the request, the Hosting CSE should perform the following procedures at that time and shall not perform the procedures before that time.

The Hosting CSE shall check to see if the target resource has a child <subscription> resource with *notificationEventType* attribute set to “Blocking\_Update” according to the procedure specified in clause 7.5.1.2.19.

Attributes that are not included in the ***Content*** parameter of the addressed resource shall not be changed by the Hosting CSE. For attributes provided in the ***Content*** parameter, their content shall be updated while the following rules apply:

If the *announceTo* attribute or *announcedAttribute* attribute of the resource is requested to be updated, the Hosting CSE shall update the attribute as described in the "announce the resource or attribute" and "de-announce the resource or attribute" procedures as specified in the clause 7.3.3.10 and clause 7.3.3.11, respectively.

The Hosting CSE shall check if the update causes a change to a reference to an Application Entity Resource ID. If so the Hosting CSE shall send a Notify request primitive to the IN-CSE, requesting to update the entry to the <AEContactList> resource.

**O attribute for update request**

If an attribute value is provided and the value is accepted, the Hosting CSE shall use the provided value in the resource representation of the updated resource.

If the attribute is not provided, but the attribute exists in the target resource, the Hosting CSE shall leave the value of that attribute unchanged.

If this attribute is provided in the ***Content*** parameter and does not exist in the target resource, the Hosting CSE shall create such attribute with the provided value.

If this attribute is set to NULL in the ***Content*** parameter and exists in the target resource, the Hosting CSE shall delete such attribute.

If the *expirationTime* attribute is present and modified by the procedure and it is set to a non-negative time, then an expiration timer shall be re-started by the Hosting CSE. At timer expiration the related resource is deleted as specified in "Delete the addressed resource".

**NP attribute for update request**

If the update is successful, the Hosting CSE shall set the *lastModifiedTime* to the current time and the Hosting CSE shall increment the *stateTag* if present.

### -----------------------End of change 7---------------------------------------------

CHECK LIST

* Does this Change Request include an informative introduction containing the problem(s) being solved, and a summary list of proposals.?
* Does this CR contain changes related to only one particular issue/problem?
* Have any mirror CRs been posted?
* Does this Change Request make **all** the changes necessary to address the issue or problem? E.g. A change impacting 5 tables should not include a proposal to change only 3 tables?Does this Change Request follow the drafting rules?
* Are all pictures editable?
* Have you checked the spelling and grammar?
* Have you used change bars for all modifications?
* Does the change include the current and surrounding clauses to clearly show where a change is located and to provide technical context of the proposed change? (Additions of complete clauses need not show surrounding clauses as long as the proposed clause number clearly shows where the new clause is proposed to be located.)
* Are multiple changes in this CR clearly separated by horizontal lines with embedded text such as, start of change 1, end of change 1, start of new clause, end of new clause.?