|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* | ARC 31 |
| Source:\* | Bob Flynn, Convida Wireless, [Flynn.Bob@Convidawireless.com](mailto:Flynn.Bob@Convidawireless.com)  Dale Seed, Convida Wireless, [Seed.Dale@Convidawireless.com](mailto:Seed.Dale@Convidawireless.com)  Bhargavi Rao, Convida wireless, [BhargaviNagarajaRao.Chanakesapura@Convidawireless.com](mailto:BhargaviNagarajaRao.Chanakesapura@Convidawireless.com) |
| Date:\* | 2017-09-21 |
|  |  |
| Reason for Change/s:\* | Provide clarification for how to trigger “storage based” events |
| CR against: Release\* | Release 3 |
| CR against: WI\* | Active <Work Item number>  MNT maintenance / < Work Item number(optional)>  Is this a companion CR? Yes  No  Companion CR number: (Note to Rapporteur - use latest agreed revision)  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\* | TS-0001V3.7.0 |
| Clauses \* | Section 9.6.24, 9.6.35, 9.6.37, 10.2.11.6 |
| 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) | TS-0004 |
| 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.

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 sections need not show surrounding clauses as long as the proposed section number clearly shows where the new section 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

R01 – Address comments during review

R02 – Rebaseline to TS-0001 v3.7.0

R03 – Add transaction back to resource types.

Discuss if we should make it R2 as well. There are compliance test implications if we do not. Further discussion needed.

The <**eventConfig**> resource defines *eventType* as “This attribute indicates the type of the event, such as timer based, data operation, storage based, etc..”

Change 1 changes the text be be more concrete since there is no mechanism to define “etc”

The dataSize attribute is defined as “This attribute defines the data size if an event is triggered when the stored data exceeds a certain size”.

Change 1 also rewords this for clarity an proper grammar.

Since the storage based eventType “is triggered when the stored data exceeds a certain size” we need to define what this means for <**containers**>, <**flexContainers**> and <**timeSeries**> as well as the corresponding <**contentInstance**> and <**timeSeriesInstance**>.

The following is proposed:

For <**containers**> *currentByteSize* is compared to the *dataSize* attribute of <**eventConfig**> - Change 1

For <**contentInstance**> *contentSize* is compared to the *dataSize* attribute of <**eventConfig**> - Change 1

For <**flexContainers**> *contentSize* is defined similar to a <**contentInstance**> as “The sum of the size in bytes of each *[customAttribute]* present in the <**flexContainer**> - Change 2

For <**flexContainers**> *contentSize* is compared to the *dataSize* attribute of <**eventConfig**> - Change 1

For <**timeSeries**> *currentByteSize* is compared to the *dataSize* attribute of <**eventConfig**> - Change 1

For <**timeSeriesInstance**> *contentSize* is defined similar to a <**contentInstance**> as “Size in bytes of the *content* attribute”. - Change 3

For <**timeSeriesInstance**> *contentSize* is compared to the *dataSize* attribute of <**eventConfig**> - Change 1

The Receiver shall verify that the *dataSize* attribute is present and contains a value greater to zero if the *eventType* is set to "Storage based" - Change 4 rewords this for grammar and clarity.

The current wording does not describe handling of eventResourceTypes and eventResourceIDs if both the attributes are present in the request. Three options are OR the values, AND the values or only allow 1 or the other to be present.

Option 1:

An OR operation is performed for monitoring these events. for example, if eventResourceTypes contains AE and eventResourceIDs contains a container resourceID, events will be triggered for both AE operations and operations on the container resourceID configured.

But a cons to this approach is if eventResourceType contains AE and eventResourceIDs contains AE resourceID, then events will be triggered for all AE operations including the specific AE resourceID configured which makes the value in eventResourceIDs redundant.

This makes the eventResourceID irrelevant. – NOT RECOMMENDED

Option 2:

An AND operation is performed for monitoring these events. For example, if eventResourceTypes contains AE and eventResourceIDs contains a container resourceID, events will not be triggered for both AE operations and operations on container resourceID configured.

Any storage based operation will not trigger this event. One way would be to validate the values of the eventResourceIDs and eventResourceTypes to check if both do not point to the same resourceTypes then return an error. This should be performed during create/update of the <eventCOnfig> resource.

This approach makes the eventResourceType act like a validation step – NOT RECOMMENDED

Option 3:

Specify these two attributes as mutally exclusive, i.e. the resource can define either eventResourceType OR eventResourceID BUT NOT BOTH – This is what is proposed in change 1.

An editorial change below uses this definition from TS-0001, 9.6.1.1

Among the resource types listed in Table 9.6.1.1-1, the following are termed "Content Sharing Resources" in oneM2M Specifications for the purpose of referring to any of those resource types:

* *container;*
* *contentInstance;*
* *flexContainer;*
* *timeSeries;*
* *timeSeriesInstance.*

## ----------------------- Start of change 1 in TS0001 -----------------------

### 9.6.24 Resource Type *eventConfig*

*<eventConfig>* sub-resource shall be used to define events that trigger statistics collection. Below are some examples of events that can be generated:

* Collection based on a certain operation: collects any RETRIEVE operations on the data (i.e. resources) stored in the IN-CSE.
* Collection based on storage size: collects the size of storage when a "Content Sharing Resource" stored in the IN-CSE exceeds a quota.
* Combined configuration: collects all RETRIEVE operations on the data stored in the IN-CSE during a period of time.

<

eventConfig

>

1

eventID

1

eventType

0

..

1

eventStart

0

..

1

eventEnd

0

..

1

dataSize

0

..

1

(

L

)

operation

Type

<

>

0

..

n

0

..

1

locationRestriction

0

..

1

(

L

)

eventResourceTypes

0

..

1

(

L

)

eventResourceIDs

<transaction>

0..n

subscription

Figure 9.6.24-1: Structure of *<eventConfig>* resource

The *<eventConfig>* resource shall contain the child resource specified in table 9.6.24-1.

Table 9.6.24-1: Child resources of *<eventConfig>* resource

| Child Resources of *<eventConfig>* | Child Resource Type | Multiplicity | Description |
| --- | --- | --- | --- |
| *[variable]* | *<subscription>* | 0..n | See clause 9.6.8 where this type of resource is described. |
| *[variable]* | *<transaction>* | 0..n | See clause 9.6.48 |

The *<eventConfig>* resource shall contain the attributes specified in table 9.6.24-2.

Table 9.6.24-2: Attributes of *<eventConfig>* resource

| Attributes of *<eventConfig>* | 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. |
| *accessControlPolicyIDs* | 0..1 (L) | RW | See clause 9.6.1.3. |
| *creationTime* | 1 | RO | See clause 9.6.1.3. |
| *expirationTime* | 1 | RW | 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. |
| *dynamicAuthorizationConsultationIDs* | 0..1 (L) | RW | See clause 9.6.1.3. |
| *creator* | 0..1 | RO | See clause 9.6.1.3. |
| *eventID* | 1 | RO | This attribute uniquely identifies the event to be collected for statistics for AEs. |
| *eventType* | 1 | RW | This attribute indicates the type of the event: timer based, data operation, or storage based. |
| *eventStart* | 0..1 | RW | This attribute indicates the start time of the event. |
| *eventEnd* | 0..1 | RW | This attribute indicates the end time of the event |
| *operationType* | 0..1 (L) | RW | This attribute defines the type of the operation to be collected by statistics, such as CREATE, RETRIEVE. |
| *dataSize* | 0..1 | RW | This attribute defines the data size that will trigger a storage based event. For <container> and <timeSeries> *currentByteSize* is compared. For <contentInstance>, <flexContainer>, <timeSeriesInstance> *contentSize* is compared. An event is triggered when the compared data size exceeds *dataSize* size. |
| *eventResourceTypes* | 0..1 (L) | RW | This attribute indicates the list of resource types for which an event is to be captured and reported. This could be used to differentiate the same operation on different types of resources that triggers the charging activity. If this attribute is specified then *eventResourceIDs* shall not be specified. |
| *eventResourceIDs* | 0..1 (L) | RW | This attribute indicates the list of resourceIDs for which the event is to be captured and reported. Whenever an operation is performed on the resourceIDs in this list, an event will be recorded provided other event criterias are met such as eventResourceType, locationRestriction and the event information based on the type of event. If this attribute is specified then *eventResourceTypes* shall not be specified. |

## ----------------------- End of change 1 in TS0001-----------------------

## ----------------------- Start of change 2 in TS0001 -----------------------



### 9.6.35 Resource Type *flexContainer*

The *<flexContainer>* resource type is a customizable container for data instances. It is a template for the definition of flexible specializations of data containers. Like a <*container*> resource, specializations of this *<flexContainer>* resource type are used to share information with other entities and potentially to track the data. While the <*container*> resources includes data to be made accessible to oneM2M entities inside <*contentInstance*> children, a specialization of the *<flexContainer>* resource includes associated content directly inside the <*flexContainer*> by means of one or more [*customAttribute*] attribute(s). The attribute name and attribute data type of [*customAttribute*] attributes are defined explicitly for each specialization of <*flexContainer>*, i.e. the specific set of attribute name and type are defined in a corresponding XSD-file.

Example usage of *<flexContainer>*: As a specialization of <*flexContainer*> that includes two [customAttribute] attributes, named "temperature"(xs:float type) and "humidity"(xs:positiveInteger type) can be specified in some TS. The actual data types of [customAttribute] will be described both in the specification document or XSD file which are referred by the value of *containerDefinition* attribute.



Figure 9.6.35-1: Structure of <*flexContainer*> resource

The *<flexContainer>* resource shall contain the child resource specified in table 9.6.35-1.

Table 9.6.35-1: Child resources of <*flexContainer*> resource

| **Child Resources of <*flexContainer*>** | **Child Resource Type** | **Multiplicity** | **Description** | ***<****flexContainer****Annc>* Child Resource Type** |
| --- | --- | --- | --- | --- |
| *[variable]* | *<semanticDescriptor>* | 0..n | See clause 9.6.30 | *<semanticDescriptor>, <semanticDescriptorAnnc>* |
| *[variable]* | *<subscription>* | 0..n | See clause 9.6.8 | *<subscription>* |
| *[variable]* | *<container>* | 0..n | See clause 9.6.6 | *<container>*  *<containerAnnc>* |
| *[variable]* | *<flexContainer>* | 0..n | <flexContainer> resource can include any of its specializations as child resource | *<flexContainer>*  *<flexContainerAnnc>* |
| *[variable]* | *<timeSeries>* | 0..n | See clause 9.6.36 | *<timeSeries>,*  *<timeSeriesAnnc>* |
| *[variable]* | *<transaction>* | 0..n | See clause 9.6.48 | *<transaction>* |

The *<flexContainer>* resource shall contain the attributes specified in table 9.6.35-2.

Table 9.6.35-2: Attributes of <*flexContainer*> resource

| **Attributes of  *<flexContainer>*** | **Multiplicity** | **RW/**  **RO/**  **WO** | **Description** | ***<flexContainerAnnc>* Attributes** |
| --- | --- | --- | --- | --- |
| *resourceType* | 1 | RO | See clause 9.6.1.3. | NA |
| *resourceID* | 1 | RO | See clause 9.6.1.3. | NA |
| *resourceName* | 1 | WO | See clause 9.6.1.3. | NA |
| *parentID* | 1 | RO | See clause 9.6.1.3. | NA |
| *expirationTime* | 0..1 (note) | RW | See clause 9.6.1.3. | MA |
| *accessControlPolicyIDs* | 0..1 (L) | RW | See clause 9.6.1.3. | MA |
| *labels* | 0..1 (L) | RW | See clause 9.6.1.3. | MA |
| *creationTime* | 0..1 (note) | RO | See clause 9.6.1.3. | NA |
| *lastModifiedTime* | 0..1 (note) | RO | See clause 9.6.1.3. | NA |
| *stateTag* | 1 | RO | See clause 9.6.1.3.  This *stateTag* attribute value shall be incremented when a <*container*> or [*flexContainer*] child resource is created or deleted. This works same as the *stateTag* attribute update on a <container> resource at a <contentInstance> resource creation or deletion. | OA |
| *announceTo* | 0..1 (L) | RW | See clause 9.6.1.3. | NA |
| *announcedAttribute* | 0..1 (L) | RW | See clause 9.6.1.3. | NA |
| *dynamicAuthorizationConsultationIDs* | 0..1 (L) | RW | See clause 9.6.1.3. | OA |
| *creator* | 0..1 | RO | See clause 9.6.1.3. | NA |
| *containerDefinition* | 1 | WO | This contains an identifier reference (URI) to the <*flexContainer*> schema definition which shall be used by the CSE to validate the syntax of the <*flexContainer*> resource.  This URI may refer to one of the oneM2M <*flexContainer*> defintions specified in the following documents:   * Generic Interworking [6]] * AllJoyn Interworking [7]; * Home Domain Information Model [8]   A list of oneM2M <*flexContainer*> defintions is also provided in clause 9.6.1.2.2 [3].  Other URI for other *<flexContainer>* definitions may be specified. | MA |
| *ontologyRef* | 0..1 | RW | A reference (URI) of the ontology used to represent the information that is stored in the present *<flexContainer>* resource. | OA |
| *contentSize* | 1 | RO | Sum of the size in bytes of all of the custom attributes. | OA |
| *[customAttribute]* | 0..n | RW | Specialization-specific attribute(s). Name and data type defined in each specialization of <*flexContainer>* resource. | OA |
| NOTE: When an instance of <*flexContainer*> is a child of a <*flexContainer*> resource, these attributes can be optional. Their presence is determined by the respective definition referred to by the *containerDefinition* attribute. | | | | |

## ----------------------- End of change 2 in TS0001-----------------------

## ----------------------- Start of change 3 in TS0001 -----------------------

### 9.6.37 Resource Type *timeSeriesInstance*

The *<timeSeriesInstance>* resource represents a data instance in the *<timeSeries>* resource. The *<timeSeriesInstance>* resource shall not be modified once created. An AE shall be able to delete a *<timeSeriesInstance>* resource explicitly or it may be deleted by the platform based on policies. If the platform has policies for *<timeSeriesInstance>* retention, these shall be represented by the attributes *maxByteSize*, *maxNrOfInstances* and/or *maxInstanceAge* attributes in the *<timeSeries>* resource. If multiple policies are in effect, the strictest policy shall apply. The *<timeSeriesInstance>* resource inherits the same access control policies of the parent *<timeSeries>* resource, and does not have its own *accessControlPolicyIDs* attribute.



Figure 9.6.37-1: Structure of <*timeSeriesInstance*> resource

Table 9.6.37-1:Child resources of <*timeSeriesInstance*> resource

| Child Resources of *<timeSeriesInstance>* | Child Resource Type | Multiplicity | Description | *<timeSeriesInstanceAnnc>* Child Resource Types |
| --- | --- | --- | --- | --- |
| *[variable]* | *<transaction>* | 0..n | See clause 9.6.48 | *<transaction>* |

The < *timeSeriesInstance*> resource shall contain the attributes specified in table 9.6.37-2.

Table 9.6.37-2: Attributes of <*timeSeriesInstance*> resource

| Attributes of *<timeSeriesInstance>* | Multiplicity | RW/  RO/  WO | Description | *<timeSeriesInstanceAnnc>* Attributes |
| --- | --- | --- | --- | --- |
| *resourceType* | 1 | RO | See clause 9.6.1.3. | NA |
| *resourceID* | 1 | RO | See clause 9.6.1.3. | NA |
| *resourceName* | 1 | WO | See clause 9.6.1.3. | NA |
| *parentID* | 1 | RO | See clause 9.6.1.3. | NA |
| *labels* | 0..1 (L) | WO | See clause 9.6.1.3. | MA |
| *creationTime* | 1 | RO | See clause 9.6.1.3. | NA |
| *expirationTime* | 1 | WO | See clause 9.6.1.3. | NA |
| *announceTo* | 0..1 (L) | RW | See clause 9.6.1.3. | NA |
| *announcedAttribute* | 0..1 (L) | RW | See clause 9.6.1.3. | NA |
| *lastModifiedTime* | 1 | RO | See clause 9.6.1.3. | NA |
| *dataGenerationTime* | 1 | WO | This attribute contains the time when the data was generated by the AE/CSE. | OA |
| *content* | 1 | WO | This attribute contains the data generated by the AE/CSE. | OA |
| *contentSize* | 1 | RO | Size in bytes of the *content* attribute. | OA |
| *sequenceNr* | 0..1 | WO | This attribute contains the data sequence number generated by the AE/CSE | OA |

## ----------------------- End of change 3 in TS0001-----------------------

## ----------------------- Start of change 4 in TS0001 -----------------------

#### 10.2.11.6 Create *<eventConfig>*

This procedure shall be used to create *<eventConfig>* resource.

Table 10.2.11.6-1: *<eventConfig>* CREATE

|  |  |
| --- | --- |
| *<eventConfig>* CREATE | |
| Associated Reference Points | Mca |
| Information in Request message | ***From:*** Identifier of the AE that initiates the Request  ***To:*** The address of the *<statsConfig>* resource where the *<eventConfig>* sub‑resource is intended to be Created  ***Content*:** The representation of the *<eventConfig>* resource for which the attributes are described in clause 9.6.24  Other information in the Request message is defined according to clause 10.1.2 |
| Processing at Originator before sending Request | The Originator shall be an AE. The Originator shall request to Create a new *<eventConfig>* resource by addressing to the *<statsConfig>* resource of a Hosting CSE |
| Processing at Receiver | The Receiver shall be an IN-CSE:   * The Receiver shall check if the *eventID* is unique, and if not, provides a new value * The Receiver shall verify that the *eventEnd* time is greater than the *eventStart* time if these two attributes are present * The Receiver shall verify that the *dataSize* attribute is present and contains a value greater than zero if the *eventType* is set to "Storage based" |
| Information in Response message | According to clause 10.1.2 |
| Processing at Originator after receiving Response | None |
| Exceptions | According to clause 10.1.2 |

## ----------------------- End of change 4 in TS0001-----------------------

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?
* Does this change request make **all** the changes necessary to address the issue or problem? 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?
* 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 sections need not show surrounding clauses as long as the proposed section number clearly shows where the new section 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.?