|  |
| --- |
|  |

|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* | SDS 54 |
| Source:\* | Bob Flynn, Exacta, [bob.flynn@exactagss.com](mailto:bob.flynn@exactagss.com)  Miguel Angel Reina Ortega, ETSI, [MiguelAngel.ReinaOrtega@etsi.org](mailto:MiguelAngel.ReinaOrtega@etsi.org) |
| Date:\* | 2021-02-04 |
| Reason for Change/s:\* | Latest-Oldest multiplicity |
| CR against: Release\* | Rel-4 |
| CR against: WI\* | Active < WI-0077>  MNT maintenance / < Work Item number(optional)>  Is this a mirror CR? Yes  No  mirror CR number: SDS-2021-0049R01-TS-0001\_latest\_oldest\_multiplicity\_R3  STE Small Technical Enhancements / < Work Item number (optional)>  Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0001 v4.8.0 |
| Clauses \* | 9.6.6, 9.6.36, 8.1.2, 9.5.0 |
| 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 |
| Other TS/TR(s) impacted | None |
| 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 2019 (do not modify) | |

**oneM2M Notice**

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

GUIDELINES for Change Requests:

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

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

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

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

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

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

Follow the drafting rules.

All pictures must be editable.

Check spelling and grammar to the extent practicable.

Use Change bars for modifications.

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

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

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

## Introduction

This CR proposes an editorial change for multiplicity of latest/oldest virtual resources in Container and TimeSeries.

R01 – Clarification on the meaning of multiplicity for virtual resources.

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

**Table 9.6.6-1: Child resources of *<container>* resource**

| **Child Resources of *<container>*** | **Child Resource Type** | **Multiplicity** | **Description** | ***<containerAnnc>* Child Resource Types** |
| --- | --- | --- | --- | --- |
| *[variable]* | *<semanticDescriptor>* | 0..n | See clause 9.6.30 | *<semanticDescriptor>, <semanticDescriptorAnnc>* |
| *[variable]* | *<contentInstance>* | 0..n | See clause 9.6.7 | *<contentInstance>, <contentInstanceAnnc>* |
| *[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 | See clause 9.6.35 | *<flexContainer>*  *<flexContainerAnnc>* |
| *[variable]* | *<timeSeries>* | 0..n | See clause 9.6.36 | *<timeSeries>,*  *<timeSeriesAnnc>* |
| *la* | *<latest>* | 1 | See clause 9.6.27 | *None* |
| *ol* | *<oldest>* | 1 | See clause 9.6.28 | *None* |
| *[variable]* | *<transaction>* | 0..n | See clause 9.6.48 | *<transaction>* |
| *[variable]* | *<action>* | 0..n | See clause 9.6.61 | *None* |

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

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

**Table 9.6.36-1: Child resources of <*timeSeries*> resource**

| **Child Resources of *<timeSeries>*** | **Child Resource Type** | **Multiplicity** | **Description** | ***<timeSeriesAnnc>* Child Resource Types** |
| --- | --- | --- | --- | --- |
| *[variable]* | *<semanticDescriptor>* | 0..n | See clause 9.6.30 | *<semanticDescriptor>, <semanticDescriptorAnnc>* |
| *[variable]* | *<timeSeriesInstance>* | 0..n | See clause 9.6.37 | *<timeSeriesInstance>,*  *<timeSeriesInstanceAnnc>* |
| *[variable]* | *<subscription>* | 0..n | See clause 9.6.8 | *<subscription>* |
| *la* | *<latest>* | 1 | See clause 9.6.27 | *None* |
| *ol* | *<oldest>* | 1 | See clause 9.6.28 | *None* |
| *[variable]* | *<transaction>* | 0..n | See clause 9.6.48 | *<transaction>* |
| *[variable]* | *<action>* | 0..n | See clause 9.6.61 | *None* |

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

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

* ***Result Content*:** optional result content: Indicates what are the expected components of the result of the requested operation. This shall be indicated in the ***Content*** parameter of the response message. Settings of ***Result Content*** depend on the requested operation specified in ***Operation***. This parameter is not applicable when ***Response Type*** has a value of *No Response*. Possible values of ***Result Content*** are:
* **attributes:** A representation of the targeted resource including all its attributes shall be returned as content, without the address(es) of the child resource(s) or their descendants. For example, if the request is to retrieve a *<container>* resource, the address(es) of the *<contentInstance>* child-resource(s) is not provided. This setting shall be only valid for Create, Retrieve, Update, or Delete operation. If the Originator does not set ***Result Content*** parameter in a Create, Retrieve or Update request message, this setting shall be the default value when the Receiver processes the request message.
* **modified-attributes**: This setting shall be only valid for a Create or Update operation. A representation of the targeted resource including only attributes that were added, modified or deleted that were not included in the request ***Content*** parameter as well as any attributes which were set to values different from the values specified in the request ***Content*** parameter shall be returned as content, without the address(es) of the child resource(s) or their descendants.
* **hierarchical-address:** Representation of the address of the created resource. This setting shall only be valid for a Create operation. The address shall be in hierarchical address scheme.
* **hierarchical-address+attributes:** Representation of the address in hierarchical address scheme and the attributes of the created resource. This setting shall only be valid for a Create operation.

- **attributes+child-resources:** Representation of the requested resource, along with a nested representation of all of its child resource(s), and their descendants (excluding virtual resources), in line with any provided filter criteria as given in the ***Filter Criteria*** parameter shall be returned as content. If there is no filter criteria parameter in the request message, then all children/descendants are returned along with their attributes. For example, if the request is to retrieve a *<container>* resource that only has *<contentInstance>* children, the attributes of that *<container>* resource and a representation of all of its *<contentInstance>* child-resource(s), including their attributes, are provided.

The originator may request to limit the maximum number of allowed nesting levels. The originator may also include an offset that indicates the starting point of the direct child resource. The offset shall start at 1. The hosting CSE shall return all direct child resources and their descendants, or up to the maximum nesting level specified in a request subject to maximum size limit that may be imposed by the hosting CSE. The offset, maximum number/size and maximum level shall be specified in ***Filter Criteria*** as *offset*, *limit*, and *level* condition, respectively, by the Originator.

The hosting CSE shall list parent resources before their children. This means that the originator of the request will not receive a descendant resource without having received its parents. The hosting CSE shall also ensure that proper nesting representation of all the children is incorporated in its listing for parents and children.

Nested processing is applicable at every level in the resource tree. If a direct child resource and all its descendants cannot be included in the returned content due to size limitations imposed by the hosting CSE then the direct child resource shall not be included in the response.

An indication shall be included in the response signalling if the returned content is partial. If the indication is for partial content, the response shall include an offset for the direct child resource where processing can restart for the remaining direct child resources

This shall be only valid for a Retrieve/Delete operation.

* **child-resources:** A nested representation of the resource's child resource(s) their descendants (excluding virtual resources) and their attributes shall be returned as content. The resources that are returned are subject to any filter criteria that are given in the ***Filter Criteria*** parameter (if there are no filter criteria then all children and their descendants are returned). The attributes of the parent resource are not returned, but all the attributes of the children are returned. For example, if the request is to retrieve a *<container>* resource that only has *<contentInstance>* children, only a representation of all of its *<contentInstance>* child-resource(s) is provided.

The offset, maximum number/size and maximum level shall be specified in ***Filter Criteria*** as *offset*, *limit*, and *level* condition, respectively, by the Originator. Processing of direct child resources, size limitations, maximum nesting level, and offset for the starting of direct child resource processing of **the attributes+child-resources** option shall apply to this option as well.

This shall be only valid for a Retrieve/Delete operation.

* **attributes+child-resource-references:** Representation of the requested resource, along with the address(es) of the child resource(s), and their descendants (excluding virtual resources) shall be returned as content. For example, if the request is to retrieve a *<container>* resource, the *<container>* resource and the address(es) of the *<contentInstance>* child-resource(s) are provided.

The offset, maximum number/size and maximum level shall be specified in ***Filter Criteria*** as *offset*, *limit*, and *level* condition, respectively, by the Originator. Processing of child resources, size limitations, maximum nesting level, and offset for the starting of child resource processing of **the attributes+child-resources** option shall apply to this option as well.

This shall be only valid for a Retrieve/Delete operation.

* **child-resource-references:** Address(es) of the child resources and their descendants (excluding virtual resources), without any representation of the actual requested resource shall be returned as content. For example, if the request is to retrieve a *<container>* resource, only the address(es) of the *<contentInstance>* child-resource(s) is provided.

The offset, maximum number/size and maximum level shall be specified in ***Filter Criteria*** as *offset*, *limit*, and *level* condition, respectively, by the Originator. Processing of child resources, size limitations, maximum nesting level, and offset for the starting of child resource processing of **the attributes+child-resources** option shall apply to this option as well.

This shall be only valid for a Retrieve/Delete operation.

This option can be used within the context of resource discovery mechanisms (see clause 10.2.6).

* **nothing:** Nothing shall be returned as operational result content. If the Originator does not set the ***Result Content*** parameter in a Delete request message, this setting shall be the default value when the Receiver processes the request message. This setting shall be valid for a Create, Update, Delete, or Notify operation.

EXAMPLE: If the request is to delete a resource, this setting indicates that the response shall not include any content.

* **original-resource:** Representation of the original resource pointed by the *link* attribute in the announced resource shall be returned as content, without the address(es) of the child resource(s). This shall be only valid for a Retrieve operation where the ***To*** parameter targets the announced resource.
* **semantic-content:** Representation of semantic information that is the result of a semantic query as indicated by the setting of the ***Semantic Query Indicator*** parameter.

This shall be only valid for a Retrieve operation with Semantic Query Indicator parameter set.

* **discovery-result-references:** For Discovery or IPE On-demand Discovery requests address(es) of discovered resources. For Discovery-based operations address(es) of discovered resources where the requested operation was performed successfully or unsuccessfully (see clause 10.2.6).

This shall be only valid when the *filterUsage* condition of the ***Filter Criteria*** parameter is set to ‘discovery’, 'discoveryBasedOperation' or 'IPEOnDemandDiscovery'.

* **permissions**: Representation of the privileges that the Originator has for the targeted resource(s). The result is a consolidated representation of all the privileges defined in the <accessControlPolicy> resources associated with the targeted resource(s), that have applicability to the Originator, and that the Originator has privileges to access (i.e. Originator has RETRIEVE *selfPrivileges* for the <accessControlPolicy> resources).

Table 8.1.2-1: Summary of Result Content Values

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Value** | **Create** | **Retrieve** | **Update** | **Delete** | **Notify** | **Retrieve**  **(filterUsage=’discovery’)** |
| attributes | default | default | default | valid | n/a | n/a |
| modified-attributes | valid | n/a | valid | n/a | n/a | n/a |
| hierarchical-address | valid | n/a | n/a | n/a | n/a | n/a |
| hierarchical-address+attributes | valid | n/a | n/a | n/a | n/a | n/a |
| attributes+child-resources | n/a | valid | n/a | valid | n/a | n/a |
| child-resources | n/a | valid | n/a | valid | n/a | n/a |
| attributes+child-resource-references | n/a | valid | n/a | valid | n/a | n/a |
| child-resource-references | n/a | valid | n/a | valid | n/a | valid |
| nothing | valid | n/a | valid | default | valid | n/a |
| original-resource | n/a | valid | n/a | n/a | n/a | n/a |
| semantic-content | n/a | valid | n/a | n/a | n/a | n/a |
| discovery-result-references | valid\* | valid\* | valid\* | valid\* | n/a | default\* |
| permissions | n/a | valid | n/a | n/a | n/a | n/a |

\*Note: additional conditions apply, see text descriptions.

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

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

### 9.5.0 Overview

The following conventions are used for the specification of resources.

Resources are specified via a tabular notation and the associated graphical representation as follows:

* The resources are specified in association with a CSE. The resources are the representation in the CSE of the components and elements within the oneM2M System. Other CSEs, AEs, application data representing sensors, commands, etc. are known to the CSE by means of their resource representation. Resource, Child Resource and Attributes are defined in clause 3.1 and are restated below for readability.
* **Resource:** A Resource is a uniquely addressable entity in oneM2M architecture. A resource is transferred and manipulated using CRUD operations (see clause 10.1). A resource can contain child resource(s) and attribute(s).
* **Child Resource:** A sub-resource of another resource that is its parent resource. The parent resource contains references to the child resources(s).
* **Attribute:** Stores information pertaining to the resource itself.
* The set of attributes, which are common to all resources, are not detailed in the graphical representation of a resource.
* Resource names and attribute names are strings in lower case. In case of a composed name, the subsequent word(s) start with a capital letter; e.g. *accessControlPolicy*, *creationTime*, *expirationTime*.
* Resource type names and attribute names are written in *italic* form in the present document.
* A string containing resource type name in *italic* delimited with '<' and '>' e.g. *<resourceType>* is used as an abbreviation referring to the type of a resource. For example, the text "a *<container>* resource" could be used as an abbreviation for "a resource of type *container*".
* A string containing a resource type name delimited with '[' and ']' e.g. *[resourceType]* is an abbreviation referring to a specialization of a resource type.
* Specialization of a resource type is done by defining specific names and descriptions of the attributes that can be specialized from the base resource type. For example, the text "a [*battery*] resource" could be used as an abbreviation for "a resource of type *battery*", where battery is a specialization of base resource type *mgmtObj*.
* A string containing an attribute type name in italic delimited with '[' and ']', e.g. *[objectAttribute]* is used as an abbreviation referring to a type of an attribute that can be specialized. Attributes that can be specialized only occur in resource types that can be specialized.

The resources are specified as shown in figure 9.5.0-1.



Figure 9.5.0-1: *<resourceType>* representation convention

The resource specification provides the graphical representation for the resource as in figure 9.5.0-1. The graphical representation of a resource shows the multiplicity of the attributes and child resources. The set of attributes, which are common to all resources are not detailed in the graphical representation of a resource. The following graphical representations are used for representing the attributes and child resources:

* Square boxes are used for the resources;
* Square boxes with round corners are used for attributes.

Child resources in a *<resourceType>* are detailed as shown in table 9.5.0-1.

The child resource table for an announce-able *<resourceType>* resource includes an additional column titled '*<resourceTypeAnnc>* Child Resource Types', indicating the type of announced resources. See clause 9.6.26 for further details.

An announced resource may have child resources, and such child resources can be of type "normal" or "announced". Child resources are of type "announced" when the child resources are announced independently of the original resource, as needed by the resource announcing CSE. Child resources are of type "normal" when child resources at the announced resource are created locally by the remote CSE.

Table 9.5.0-1: Child Resources of *<resourceType>*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Child Resources of *<resourceType>* | Child Resource Type | Multiplicity | Description | *<resourceTypeAnnc>*  Child Resource Types |
| <Fill in the name of Child Resource1 if a fixed name is required or [variable] if no fixed name is required> | <Fill in the type of Child Resource1> | <Fill in Multiplicity> | See clause <XRef> <clause> where the type of this child resource is described. | <Fill the child resource type for the announced resource. It can be none or *<crTypeAnnc>* or *<crType>*; where the *<crType>* is the child resource type of the original Child Resource1. |
| <Fill in the name of Child ResourceN if a fixed name is required or [variable] if no fixed name is required> | <Fill in the type of Child ResourceN> | <Fill in Multiplicity> | See clause <XRef> <clause> where the type of this child resource is described. | <Fill the child resource type for the announced resource. It can be none or *<crTypeAnnc>* or *<crType>;* where the *<crType>* is the child resource type of the original Child ResourceN. |

Attributes in a *<resourceType>* are detailed as shown in table 9.5-2.

The attributes table for announce-able *<resourceType>* resource includes an additional column titled 'Attributes for *<resourceTypeAnnc>*', indicating the attributes that are to be announced for that *<resourceType>.* See the clause 9.6.26 for further details.

Table 9.5.0-2: Attributes of *<resourceType>* resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attributes of *<resourceType>* | Multiplicity | RW/  RO/  WO | Description | *<resourceTypeAnnc>* (MA/OA/NA) |
| <Fill in name of Common Attribute1> | <Fill in Multiplicity> | <Fill in RW or RO or WO> | Provide description of this attribute - to be moved later to a common attribute clause. | <Fill in MA or OA or NA> |
| <Fill in name of Common AttributeN> | <Fill in Multiplicity> | <Fill in RW or RO or WO> | Provide description of this attribute - to be moved later to a common attribute clause. | <Fill in MA or OA or NA> |
| <Fill in name of Resource Specific Attribute1> | <Fill in Multiplicity> | <Fill in RW or RO or WO> | Provide description of this attribute - to be moved later to a central attribute table that also defines the type of the attribute, allowed ranges, etc. | <Fill in MA or OA or NA> |
| <Fill in name of Resource-Specific AttributeN> | <Fill in Multiplicity> | <Fill in RW or RO or WO> | Provide description of this attribute - to be moved later to a central attribute table that also defines the type of the attribute, allowed ranges, etc. | <Fill in MA or OA or NA> |

In case of misalignment of the graphical representation of a resource and the associated tabular representation, tabular representation shall take precedence.

The access modes for *attributes* can assume the following values:

* Read/Write (RW): the value of the attribute is set when the resource is Created or Updated based on information from the Originator (i.e. ***Content*** parameter). Such attributes are allowed for Create/Update/Retrieve operations. Note that such an attribute can be deleted by Update operation.
* Read Only (RO): the value of the attribute is set or can be updated by the Hosting CSE internally. Such an attribute is allowed for Retrieve operation only.
* Write Once (WO): the value of the attribute is set when the resource is Created based on information from the Originator (i.e. ***Content*** parameter). Such an attribute is allowed for Retrieve operation after the creation. Such attribute can thereafter only be updated by hosting CSE internally.

The multiplicity, both for the child resources and the attributes can have the following values:

* A value of "0" indicates that the child resource/attribute shall not be present.
* A value of "1" indicates that the child resource/attribute shall be present.
* A value of "0..1" indicates that the child resource/attribute may be present.
* A value of "0..n" indicates that the child resource/attribute may be present. If present, multiple instances are supported.
* A value of "1..n" indicates that the child resource shall always be present. It has at least one instance and can have multiple instances.
* An attribute multiplicity post-fixed with (L) indicates that it is a list of values.

The multiplicity for the virtual child resources can have the following values:

A value of "1" indicates that the functionality related to the virtual child resource shall be supported.

A value of "0..1" indicates that the functionality related to the virtual child resource may be supported.

The attributes for *<resourceTypeAnnc>* in the attribute table can have the following set of values:

* **MA** (Mandatory Announced): The attribute in the original resource is announced to the announced resource. The content of such an announced attribute is the same as the content of the original attribute.
* **OA** (Optional Announced): The attribute in the original resource may be announced to the announced resource depending on the contents of the *announcedAttribute* attribute at the original resource. The content of such an announced attribute is the same as the content of the original attribute.

**NA** (Not Announced): The original attribute is not announced to the announced resource.

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