|  |
| --- |
|  |

|  |
| --- |
| CHANGE REQUEST |
| Meeting ID:\* |  SDS #48 |
| Source:\* | Andreas Kraft, DT, Andreas.Kraft@t-systems.com Andreas Neubacher, DT, Andreas.Neubacher@magenta.at  |
| Date:\* | 2020-09-14 |
| Reason for Change/s:\* | Adding missing URL encodings for parent and child attributes to TS-0009 (R4) |
| CR against: Release\* | Release 4 |
| 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-0009 v.4.0.0 |
| Clauses \* | Modified clauses: 6.2.2.2 |
| Type of change: \* | [ ]  Editorial change[x]  Bug Fix or Correction[ ]  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 CR is a mirror CR for SDS-2020-270 for R4.

TS-0001 table 8.1.2-2 “Filter Criteria Conditions” define two condition tags, ‘childAttribute’ and ‘parentAttribute’, which, similar to the ‘attribute’ condition defined in the same table, define filter conditions to match any attribute in a resource’s child resp parent resorcures. Further details on the processing is given in TS-0004 clause 7.3.3.17.9.

The http protocol binding in TS-0009 specifies the mapping of “attribute” conditions but leaves out similar mappings for ‘childAttribute’ and ‘parentAttribute’ conditions. This CR proposes a change to the appropriate section in TS-0009 to also cover those filter conditions.

The solution proposes to use prefixes to distinguish the different targets: ‘c.’ for distinguishing child resource attribute conditions, and ‘p.’ for distinguishing parent resource attributes conditions. The advantages are the minimum extra processing on the CSE to handle the prefixes as well as that it is aligned with the normal ‘attribute’ scheme. The disadvantage is that care must be taken to avoid overlapping with attributes starting with one of the prefixes.

Another solution would be to pack all attribute conditions that do not belong to the target resource into a JSON (or similar) structure. This would avoid the problem of the proposed solution (ie. overlapping of attribute names starting with a prefix string), but would introduce a whole new inner format to pack the attributes.

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of change 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### 6.2.2.2 Query component

The query component (e.g. query-string) may include the optional primitive parameters listed in table 6.2.2.2-1 compliant with IETF RFC 7230 [1]. Each applicable request primitive parameters and elements of ***Filter Criteria*** parameter shown in table 6.2.2.2-1 shall be represented as pair of field-name and value in query-string. Multiple such pairs shall be concatenated with an ampersand ‘&’ character used as separator between two pairs.

Table 6.2.2.2-1 also shows the permitted multiplicity of occurrence of field names in the query-string. Multiplicity ‘0..1’ means that a parameter is optional and can occur at most once. Parameters with multiplicity ‘0..n’, may occur multiple times in the query-string in the form of <query field name> = value. For example, if the resourceType element of the ***Filter Criteria*** parameter is represented by a list of 3 values ‘2 3 4’ (see clause 6.3.4.7 in TS-0004 [3]), it would be mapped to ty=2+3+4 in the query-string. At the receiver side, this query string can be reverted back into the list type of representation. The same representation shall be applied for multiple occurrences of contentType and labels elements.

The ‘attribute’ element of the ***Filter Criteria*** request primitive parameter consists of two elements, name and value, which in XML notation would look for example as follows in case of multiplicity 2 (see clause 6.2.4.8 in TS-0004 [3]):

 <attribute>

 <name>attname1</name>

 <value>attvalue1</value>

 </attribute>

 <attribute>

 <name>attname2</name>

 <value>attvalue2</value>

 </attribute>

Each name (e.g. attname1 and attname2) shall represent a valid resource attribute name of the resource types indicated in the ty field of the query-string. The sequence of attribute elements as shown in the above example will be mapped into the query-string as attname1=attvalue1&attname2=attvalue2. The attribute names (i.e. attname1 and attname2 in the above example) shall be expressed in the form of short names as defined in clause 8.2.3 of TS-0004 [3]. Note that the <attribute> tag of the XML representation is omitted in the HTTP binding.

The ‘childAttribute’ and ‘parentAttribute’ elements of the ***Filter Criteria*** request primitive are handled in a similar way to the ‘attribute’ element. Those sequences of attribute elements will be mapped in the query string by adding a prefix to each attribute name respectively: ‘c.’ for ‘childAttribute’ and ‘p.’ for ‘parentAttribute’. This results, using the example above, in the mappings as c.attname1=attvalue1&c.attname2=attvalue2 for ‘childAttribute’, and p.attname1=attvalue1&p.attname2=attvalue2 for ‘parentAttribute’.

Examples of valid Request-Target representations are the following:

**EXAMPLE 1): Request-Target for ‘nonBlockingRequestSynch’**

Primitive parameters: To: /CSE1234/RCSE78/container234 (SP-Relative-Resource-ID)

Response Type: responseType = 1 (nonBlockingRequestSynch)

Result Persistence: P1Y2M3DT10H1M0S Request-Target: /CSE1234/RCSE78/container234?rt=1&rp=P1Y2M3DT10H1M0S

**EXAMPLE 2): Request-Target for Discovery**

When the entity wants to discover container resources where the *creator* attribute has the value ‘Sam’:

Primitive parameters: To: /CSE1234/RCSE78

 Filter Criteria: resourceType = 3 (container)

 attribute name: creator

 attribute value: Sam

 filterUsage = discovery

Request-Target: /CSE1234/RCSE78?ty=3&cr=Sam&fu=1

**EXAMPLE 3): Semantic Discovery**

The entity wants to discover resources whose semantic description stored in the *descriptor* attribute of a <semanticDescriptor> child resource fulfils the semantic filter specified in SPARQL. In this case, the semantic descriptor of the resource to discover has to contain information about a Thing of type Car based on the concept defined in the “myOnt” ontology.

Due to the use of reserved characters in SPARQL, the semanticsFilter requires “percent-encoding” [9].

Primitive parameters: To: /CSE1234/RCSE78

Filter Criteria: semanticsFilter =
PREFIX rdf: [http://www.w3.org/1999/02/22-rdf-syntax-ns#](http://www.w3.org/1999/02/22-rdf-syntax-ns)
PREFIX myOnt: [http://www.onem2m.org/ontology/myontology#](http://www.onem2m.org/ontology/myontology)
SELECT ?car WHERE { ?car rdf:type myOnt:Car }

Request‑Target:        /CSE1234/RCSE78?smf=PREFIX%20rdf%3A%20%3Chttp%3A%2F%2                             Fwww.w3.org%2F1999%2F02%2F22‑rdf‑syntax‑ns%23%3E%20PREFI                                       X%20myOnt%3A%20%3Chttp%3A%2F%2Fwww.onem2m.org%2Fonto                                      logy%2Fmyontology%23%3E%20SELECT%20%3Fcar%20WHERE%20                                      %7B%20%3Fcar%20%20rdf%3Atype%20myOnt%3Acar%20%7D

**EXAMPLE 4): Geo-query**

When an application wants to query a resource having its geo-location within the rectangle:

Primitive parameters: To: /CSE1234/RCSE78

 Filter Criteria: geometryType = 3 (Polygon)

geometry = [[0.0, 0.0], [0.0, 100.0], [100.0, 100.0], [100.0, 0.0], [0.0, 0.0]]

 geoSpatialFunction = 1 (Within)

 filterUsage = discovery

Request-Target: /CSE1234/RCSE78?fu=1&gmty=3&geom=[[0.0,0.0],[0.0,100.0],[100.0,100.0],[100.0,0.0],[0.0,0.0]]&gsf=1

Note that, in the HTTP Request-Target, longitude and latitude are separated by an empty space and each pair of longitude and latitude are separated by comma.

Any of the short names listed in table 6.2.2.2-1, with the exception of ‘atr’, may be used in the query-string. The short name ‘atr’ itself is not used. Instead, any of the resource attribute short names as listed in tables 8.2.3-1 to 8.2.3-5 in oneM2M TS-0004 [3] may be used in the query-string in representations of attname=attvalue expressions, except those that shall be omitted (see clause 7.3.3.17.9 in oneM2M TS-0004 [3]).

Table 6.2.2.2-1: oneM2M request parameters mapped as query-string field

|  |  |  |  |
| --- | --- | --- | --- |
| Request Primitive Parameter | Query Field Name | Multiplicity | Note |
| Response Type | ***rt*** | 0..1 | *responseType* element of data type ***responseTypeInfo*** (cf. clause 6.3.4.29 of TS-0004 [3]) |
| Result Persistence | ***rp*** | 0..1 |  |
| Result Content | ***rcn*** | 0..1 |  |
| Delivery Aggregation | ***da*** | 0..1 |  |
| createdBefore | ***crb*** | 0..1 | filterCriteria condition |
| createdAfter | ***cra*** | 0..1 | filterCriteria condition |
| modifiedSince | ***ms*** | 0..1 | filterCriteria condition |
| unmodifiedSince | ***us*** | 0..1 | filterCriteria condition |
| stateTagSmaller | ***sts*** | 0..1 | filterCriteria condition |
| stateTagBigger | ***stb*** | 0..1 | filterCriteria condition |
| expireBefore | ***exb*** | 0..1 | filterCriteria condition |
| expireAfter | ***exa*** | 0..1 | filterCriteria condition |
| labels | ***lbl*** | 0..n | filterCriteria condition |
| resourceType | ***ty*** | 0..n | filterCriteria condition |
| sizeAbove | ***sza*** | 0..1 | filterCriteria condition |
| sizeBelow | ***szb*** | 0..1 | filterCriteria condition |
| contentType | ***cty*** | 0..n | filterCriteria condition |
| limit | ***lim*** | 0..1 | filterCriteria condition |
| attribute | ***atr*** | 0..n | filterCriteria condition |
| filterUsage | ***fu*** | 0..1 | filterCriteria condition |
| semanticsFilter | ***smf*** | 0..n | filterCriteria condition, shall use “percent-encoding” [9] where required, see example 3) |
| filterOperation | ***fo*** | 0..1 | filterCriteria condition |
| contentFilterSyntax | ***cfs*** | 0..1 | filterCriteria condition |
| contentFilterQuery | ***cfq*** | 0..1 | filterCriteria condition |
| level | ***lvl*** | 0..1 | filterCriteria condition |
| offset | ***ofst*** | 0..1 | filterCriteria condition |
| geometryType | ***gmty*** | 0..1 | filterCriteria condition |
| geometry | ***geom*** | 0..1 | filterCriteria condition |
| geoSpatialFunction | ***gsf*** | 0..1 | filterCriteria condition |
| Discovery Result Type | ***drt*** | 0..1 |  |
| Role IDs | ***rids*** | 0..n |  |
| Token IDs | ***tids*** | 0..n |  |
| LocalTokenIDs | ***ltids*** | 0..n |  |
| Token Request Indicator | ***tqi*** | 0..n |  |
| Authorization Signature Indicator | ***asi*** | 0..1 |  |
| Authorization Relationship Indicator | ***auri*** | 0..1 |  |
| Semantic Query Indicator | ***sqi*** | 0..1 |  |

For partial Retrieve request primitives, the ***To*** parameter may include the name of a single attribute separated by a ‘#’ character from the resource ID. If multiple resource attributes are to be retrieved with a partial retrieve request primitive, these attributes are included in form of an attributeList object (as specified in Table 6.3.3-1 of TS-0004 [3]) in the ***Content*** parameter.

In both cases, the short resource attribute name(s) shall be included into the fragment component of request-target, i.e. it shall follow any required query-string separated by ‘#’ character. If more than a single attribute name is included into the fragment component, these shall be separated by a ‘+’ character.

For example, if three resource attributes with long names resourceID, labels and requestReachability are indicated in the ***Content*** primitive parameter, the query component atrl=ri+lbl+rr is attached to the request-target. In case just a single attribute “rr” is indicated in the ***To*** parameter separated by ‘#’ character, the query component atrl=rr is attached to the request-target. The ‘#’ character and following attribute name shall be omitted from the path component of the request line.

Case 1 Primitive Example:

<?xml version=”1.0” encoding=”UTF-8”?>
<m2m:rqp xmlns:m2m=”http://www.onem2m.org/xml/protocols”>
 <op>1</op>
 **<to>//example.net/myCSE/-/Cont1</to>**
 <fr>/myCSE/C2345</fr>
 <rqi>0002bf63</rqi>
 <ty>4</ty>
 <pc>
 **<atrl>ri lbl rr</atrl>**
 </pc>
</m2m:rqp>

Case 1 HTTP Binding: **//example.net/myCSE/-/Cont1?atrl=ri+lbl+rr**

Case 2 Primitive Example:

<?xml version=”1.0” encoding=”UTF-8”?>
<m2m:rqp xmlns:m2m=”http://www.onem2m.org/xml/protocols”>
 <op>2</op>
 **<to>//example.net/myCSE/-/Cont1#rr</to>**
 <fr>/myCSE/C2345</fr>
 <rqi>0002bf63</rqi>
 <ty>4</ty>
</m2m:rqp>

Case 2 HTTP Binding: **//example.net/myCSE/-/Cont1?atrl=rr**

At the HTTP server side, the reverse operation shall take place, when constructing the retrieve request primitive from the receive HTTP request message. Single attribute names in the query component may either be mapped back into the ***To*** parameter following a ‘#’ character, or included into the ***Content*** parameter using the attributeList format with just a single list element included. Multiple attributes shall be included into the ***Content*** parameter as specified in oneM2M TS-0004 [3].

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