|  |
| --- |
|  |

|  |
| --- |
| CHANGE REQUEST |
| Meeting ID:\* | ARC 31 |
| Source:\* | Neeta Meshram, C-DOT, neeta@cdot.in Suman Sheoran, C-DOT, ssheoran@cdot |
| Date:\* | 2017-08-02 |
| Reason for Change/s:\* | See the introduction  |
| CR against: Release\* | Release 2 |
| CR against: WI\* | [ ]  Active <Work Item number> [x]  MNT maintenance / < Work Item number(optional)>Is this a mirror CR? Yes [ ]  No [x] 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-0001 V2\_14\_0 |
| Clauses \* | Section 9.6.12 |
| 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) | <TS/TR number>, <Version Number>, and <Description on which aspect should be reflected in this TS/TR> |
| 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 [ ]  |
| 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 proposes to change multiplicity of *primitiveContent* attribute of <request> resource from mandatory to optional.

According to TS-0001, Table 9.6.12-2, *primitiveContent* contains the content that is carried in the ***Content*** parameter of the original request message. But the request can be Retrieve and Delete request too. And Retrieve request has *primitiveContent* as optional field while Delete request do not have *primitiveContent* field. So, there is nothing to be filled in this attribute for above cases. As a solution *primitiveContent* shold be made optional.

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

### 9.6.12 Resource Type *request*

The use of *<request>* resource type is optional depending on the configuration.

Creation of a *<request>* resource can only be done on a Receiver CSE implicitly when a Registree AE or a Registree/Registrar CSE issues a request to the Receiver CSE targeting any other resource type or requesting a notification. Creation of a *<request>* resource instance is only permitted by the Receiver CSE as a result of a request from an Originator which contains the ***Response Type*** parameter in the request message and where ***Response Type*** parameter is set to *'nonBlockingReqeustSynch'* or *'nonBlockingRequestAsynch'*.

When a CSE is requested to initiate an operation for which the result should be available to the Originator by reference (***Request Expiration Timestamp*** information of the request set to *'nonBlockingReqeustSynch'* or *'nonBlockingRequestAsynch'*), the Receiver CSE which received the request directly from the Originator shall provide a reference of the created *<request>* resource back to the Originator so that the Originator can access attributes of the *<request>* at a later time - for instance in order to retrieve the result of an operation that was taking a longer time. If the Receiver CSE uses resources of type *<request>* to keep such context information, the reference that shall be given back to the Originator as part of the acknowledgment that is the address of the *<request>* resource. The Originator (or any other authorized entity depending on access control) can access the request status and the requested operation result through it.

The *<request>* resource may be deleted by the CSE that is hosting it when the expiration time of the *<request>* resource is reached. So after the expiration time of a *<request>* resource is reached it cannot be assumed that that particular *<request>* resource is still accessible. Depending on implementation of the CSE that is hosting it, a *<request>* resource may also get deleted earlier than the expiration time, when the result of the requested operation (if any result was requested at all) has been sent back to the Originator.

For the purpose of providing a standardized structure for expressing and accessing the context of a previously issued request, the resource type <*request>* is defined. The parent resource of a *<request>* resource shall be the *<CSEBase>* resource of the Hosting CSE.

<request>

1

operation

1

target

1

originator

1

requestID

1

metaInformation

0..1

primitiveContent

1

requestStatus

<subscription>

0..n

1

operationResult

Figure 9.6.12-1: Structure of *<request>* resource

The *<request>* resource shall contain the child resources specified in table 9.6.12-1.

Table 9.6.12-1: Child resources of *<request>* resource

| Child Resources of *<request>* | Child Resource Type | Multiplicity | Description |
| --- | --- | --- | --- |
| *[variable]* | *<subscription>* | 0..n | See clause 9.6.8 |

The *<request>* resource shall contain the attributes specified in table 9.6.12-2.

Table 9.6.12-2: Attributes of *<request>* resource

| Attributes of *<request>* | 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. |
| *expirationTime* | 1 | RW | See clause 9.6.1.3. The value of the *expirationTime* is chosen by the CSE dependent on the ***Request Expiration Timestamp***, ***Result Expiration Timestamp***, ***Result Persistence*** and ***Operation Execution Time*** parameters associated with the original request. |
| *parentID* | 1 | RO | See clause 9.6.1.3. |
| *creationTime* | 1 | RO | See clause 9.6.1.3. |
| *lastModifiedTime* | 1 | RO | See clause 9.6.1.3. |
| *accessControlPolicyIDs* | 0..1 (L) | RO | See clause 9.6.1.3. |
| *labels* | 0..1 (L) | RO | See clause 9.6.1.3. |
| *stateTag* | 1 | RO | See clause 9.6.1.3. |
| *dynamicAuthorizationConsultationIDs* | 0..1 (L) | RO | See clause 9.6.1.3. |
| *operation* | 1 | RO | It contains the value of the parameter ***Operation*** in the original request message. |
| *target* | 1 | RO | It contains the value of the parameter ***To*** in the original request message. |
| *originator* | 1 | RO | It contains the value of the parameter ***From*** in the original request message. |
| *requestID* | 1 | RO  | It contains the value of the parameter ***Request Identifier*** in the original request message. |
| *metaInformation* | 1 | RO | Meta information about the request. The content of this attribute is equivalent to information in any other optional parameters described in clause 8.1. |
| *primitiveContent* | 0..1 | RO | Contains the content that is carried in the ***Content*** parameter of the original request message. |
| *requestStatus* | 1 | RO | Contains information on the current status of the Request, e.g. "accepted and pending". |
| *operationResult* | 1 | RO | Contains the result of the originally requested operation in line with the ***Result Content*** parameter associated with the original request. |

All operations on *<request>* resources except for the CREATE operations - which can only be triggered implicitly by a request for which a *<request>* resource shall capture the context - are controlled by the access control policy.

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

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.?