|  |
| --- |
|  |

|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* | PRO 31 |
| Source:\* | Poornima, C-DOT, [poornima@cdot.in](mailto:poornima@cdot.in)  Suman, C-DOT, [ssheoran@cdot.in](mailto:ssheoran@cdot.in) |
| Date:\* | 2017-08-02 |
| Reason for Change/s:\* | See the introduction |
| 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\* | TS-0004 Version 3.3.0 |
| Clauses \* | 7.4.14.2.4,7.4.13.2.1,7.4.13.2.3 |
| 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/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  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

The CR is corresponding to ARC CR [ARC-2017-0355](http://member.onem2m.org/Application/documentApp/documentinfo/?documentId=23590&fromList=Y) sub-group mapping where a detail is added to distinguish sub-group members from other members. It is added that when a member is to be treated as sub-group then originator shall specify such information in memberIDs by appending /fopt in group resource identifier. So when such detail is already present in memberIDs then in TS-0004 appending /fopt will not be required.

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



##### Fanout Request to each member

If the parent group has no members, the group hosting CSE shall reject the request with the ***Response Status Code*** indicating "NO\_MEMBERS".

If the request contains a ***Group Request Target Members*** parameter, and if any of the memberIDs in this parameter is not present in original ***memberIDs*** list of group or any of the ***memberIDs*** list of sub-groups of the original group then the request shall be rejected with BAD\_REQUEST ***Response Status Code*** else the group hosting CSE shall fanout the request to members contained in this parameter only.

The group Hosting CSE shall perform the following steps for each member:

a) The primitive parameters ***From*** and ***To*** shall be mapped to the primitive parameters of the corresponding Request to be sent out to each member of the group. The primitive parameter ***From*** shall be directly used. The primitive parameter ***To*** (i.e. <URI of group resource>/fopt) shall be replaced by resource identifiers present in the ***memberIDs*** attribute of the group resource. Any additional relative address that was appended to .../fopt in the original Request shall be appended to each ***To*** URI. The group hosting CSE shall execute "Compose Request primitives". In addition, the group hosting CSE shall generate a unique group request identifier, add it as a primitive parameter to the Request and locally store the group request identifier as per the local policy.

b) "Send the Request to the receiver CSE".

c) "Wait for Response primitives".

The procedures between group hosting CSE and member hosting CSEs shall comply with the corresponding creation procedures as described in clause 7. The detailed procedures are according to the type of Resource provided in the Request primitive. During fanOutPoint manipulation, the member hosting CSE receiving a Request send from the group hosting CSE shall check if the Request contains a ***Group Request Identifier*** parameter. If the Request contains a ***Group Request Identifier*** parameter, the member hosting CSE shall compare the ***Group Request Identifier*** parameter to the ***Group Request Identifier*** locally stored. If a match is found, the member hosting CSE shall reject the request with the ***Response Status Code*** indicating "GROUP\_REQUEST\_IDENTIFIER\_EXISTS" error in the Response primitive. Otherwise, the member hosting CSE shall continue with the operations according to the Request and locally store the ***Group Request Identifier*** parameter.

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

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



##### Create

***Originator***:

For members which are of type <group>, the Originator shall suffix the ‘/fopt’ to that ‘memberID‘ during group creation if the Originator wants to fan-out the group request to each member of that sub-<group>, else the Originator shall not suffix the ‘/fopt’ to that ‘memberID‘.

***Receiver***:

Primitive specific operation after Recv-C-6.4 "Check validity of resource representation for the given resource type" and before Recv-C-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed". See clause 7.2.2.2.

1. Primitive specific operation: Validate the provided attributes. It shall also check whether the number of URIs present in the memberIDs attribute of the group resource representation does not exceed the maximum as specified by the maxNrOfMembers attribute. If the maximum is exceeded, the request shall be rejected with a ***Response Status Code*** indicating "MAX\_NUMBER\_OF\_MEMBER\_EXCEEDED" error. If there are duplicate members in the memberIDs attribute then the duplicate members are removed before creation of the <group> resource.  
   If the memberType attribute of the <group> resource is not "MIXED", the Hosting CSE shall also verify that all the member IDs including sub-groups in the attribute memberIDs of the <group> resource representation provided in the request shall conform to the memberType of the group resource. To validate a resource type of a member, the Hosting CSE shall check the *resourceType* attribute of the resource which is indicated by the member ID. To check the *resourceType* attribute, the Hosting CSE may retrieve the member resource. When a member ID is virtual resource, the Hosting CSE shall check the *resourceType* attribute of the parent resource. If the resource type of the parent allows this child virtual resource type, the Hosting CSE checks whether the virtual resource type matches with the *memberType* attribute of the group. If they match, then the Hosting CSE considers that the virtual member resource is validated.
2. In the case that the <group> resource contains sub-group member resources, the receiver shall retrieve the memberType of the sub-group member resources to validate the memberType. If the memberType cannot be retrieved due to lack of privilege, the request shall be rejected with a ***Response Status Code*** indicating "RECEIVER\_HAS\_NO\_PRIVILEGE" error. If the sub-group member resources are temporarily unreachable, the receiver shall set the memberTypeValidated attribute of the <group> resource to FALSE and return the result to the originator in the response of the request. As soon as any unreachable sub-group resource becomes reachable, the receiver shall perform the memberType validation procedure. The originator may get to know the validation result by subscribing to the created resource if the memberTypeValidated attribute is FALSE. Upon unsuccessful validation, the receiver shall delete the <group> resource if the ***consistencyStrategy*** of the <group> resource is ABANDON\_GROUP, or remove the inconsistent members from the <group> resource if the consistencyStrategy attribute is ABANDON\_MEMBER, or set the memberType attribute of the <group> resource to "MIXED" if the consistencyStrategy attribute is SET\_MIXED.
3. The *memberTypeValidated* attribute shall be set to TRUE if all the members have been validated successfully. If a member validation for the *memberType* of the <group> resource is unsuccessful, then the Hosting CSE shall perform the following:
4. If the *consistencyStrategy* of the <group> resource is ABANDON\_GROUP then the request shall be rejected with a ***Response Status Code*** indicating "GROUP\_MEMBER\_TYPE\_INCONSISTENT" error.
5. If the *consistencyStrategy* of the <group> resource is ABANDON\_ MEMBER then remove the inconsistent members and create the <group> resource and the *memberTypeValidated* attribute shall be set to TRUE.
6. If the *consistencyStrategy* of the <group> resource is SET\_ MIXED then set the *memberType* attribute of the <group> resource to "MIXED" and create the <group> resource and the *memberTypeValidated* attribute shall be set to TRUE.

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

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

##### Update

***Originator:***

If the Originator intends to update memberIDs attribute ,for members which are of type <group>, the Originator shall suffix the ‘/fopt’ to that ‘memberID‘ during group updation if the Originator wants to fan-out the group request to each member of that sub-<group> ,else the Originator shall not suffix the ‘/fopt’ to that ‘memberID‘.

***Receiver:***

1. Primitive specific operation after Recv-6.4 "Check validity of resource representation for the given resource type" and before Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed ". See clause 7.2.2.2.Primitive specific operation: If the memberType attribute of the <group> resource is not "MIXED", the Hosting CSE shall verify that all the member IDs including sub-groups in the attribute memberIDs of the <group> resource representation provided in the request shall conform to the memberType of the <group> resource. Virtual member resource validation shall be done as specified in the group creation procedure (clause 7.4.13.2.1 step 1).
2. In the case that the <group> resource contains sub-group member resources, the receiver shall retrieve the memberTypeof the sub-group member resource to validate the memberType. If the *memberType* cannot be retrieved due to lack of privilege, the request shall be rejected with a ***Response Status Code*** indicating "RECEIVER\_HAS\_NO\_PRIVILEGE" error. If the sub-group member resources are temporarily unreachable, the receiver shall set the memberTypeValidatedattribute of the <group> resource to FALSE and return the result to the originator in the response of the request. As soon as any unreachable sub-group resource becomes reachable, the receiver shall perform the *memberType* validation procedure. The Originator may get to know the validation result by subscribing to the created resource if the memberTypeValidated attribute is FALSE. Upon unsuccessful validation, the receiver shall delete the <group> resource if the consistencyStrategy of the <group> resource is ABANDON\_GROUP, or remove the inconsistent members from the<group> resource if the consistencyStrategy attribute is ABANDON\_MEMBER, or set the memberType attribute of the <group> resource to "MIXED" if the consistencyStrategy attribute is SET\_MIXED.
3. The memberTypeValidated attribute shall be set to TRUE if all the members have been validated successfully. If a member validation for the *memberType* of the <group> resource is unsuccessful, then the Hosting CSE shall perform the following:
4. If the *consistencyStrategy* of the <group> resource is ABANDON\_GROUP then the request shall be rejected with a ***Response Status Code*** indicating "GROUP\_MEMBER\_TYPE\_INCONSISTENT" error.
5. If the *consistencyStrategy* of the <group> resource is ABANDON\_ MEMBER then remove the inconsistent members and update the <group> resource and the *memberTypeValidated* attribute shall be set to TRUE.
6. If the consistencyStrategy of the <group> resource is SET\_ MIXED then set the memberType attribute of the <group> resource to "MIXED" and update the <group> resource and the memberTypeValidated attribute shall be set to TRUE.
7. Primitive specific operation: The Hosting CSE shall check whether the number of provided memberIDs in the attribute members exceeds the limitation of maxNrOfMembers. The Hosting CSE shall also check whether the value provided in the *maxNrOfMembers* attribute is smaller than the *currentNrOfMembers* attribute value. If any of the condition is true, the Hosting CSE shall reject the request with ***Response Status Code*** indicating "MAX\_NUMBER\_OF\_MEMBER\_EXCEEDED". error.

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

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.?