|  |
| --- |
|  |

|  |
| --- |
| CHANGE REQUEST |
| Meeting:\* | ARC#26.2 |
| Source:\* | C-DOT  |
| Date:\* | 2016-12-15 |
| Contact:\* | Poornima (poornima@cdot.in),Suman(ssheoran@cdot.in) ,Anupama(anupama@cdot.in)  |
| Reason for Change/s:\* | See the introduction  |
| CR against: Release\* | Release 3 |
| CR against: WI\* | [ ]  Active <Work Item number> [ ]  MNT maintenace / < Work Item number(optional)>[x]  STE Small Technical Enhancements / < Work Item number (optional)>Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0001 v2.10.1 |
| Clauses/Sub Clauses\* | 10.2.7.7,10.2.7.8,10.2.7.9,10.27.10 |
| Type of change: \* | [ ]  Editorial change[ ]  Bug Fix or Correction[ ]  Change to existing feature or functionality[x]  New feature or functionalityOnly ONE of the above shall be ticked |
| 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 [x]  NO [ ] This CR is a mirror CR? YES [ ]  if YES, please indicate the document number of the original CR: : NO [x]   |
| Template Version:27 May 2015 (Dot 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 separated “mirror CR” should be posted at the same time of this CR

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

### This CR is in continuation with Agreed CR ARC-2016-0484R02, which proposes a new parameter in request for fanout on subset of members of group.

This CR adds handling of ***groupRequestTargetMembers*** parameter in <fanoutpoint> operations

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

#### 10.2.7.7 Create *<fanOutPoint>*

This procedure shall be used for creating the content of all members resources belonging to an existing *<group>* resource.

Table 10.2.7.7-1: *<fanOutPoint>* CREATE

| *<fanOutPoint>* CREATE  |
| --- |
| Associated Reference Point | Mca, Mcc and Mcc' |
| Information in Request message | ***From:*** Identifier of the AE or the CSE that initiates the Request***To:*** The address of the *<fanOutPoint>* virtual resource***Content:*** The representation of the resource the Originator intends to create***Group Request Identifier:*** The group request identifier***Response Type:*** If the parameter is set to BlockingSynch, it indicates that the group hosting CSE shall return the aggregated response once. Otherwise if the parameter is set to nonBlockingRequestSynch, nonBlockingRequestAsynch or flexBlocking, it indicates that the Group Hosting CSE shall return the aggregated response in a batched mode***Result Expiration Time:*** Indicates the maximum time limit in which the Group Hosting CSE has to respond the aggregated response. |
| Processing at Originator before sending Request | The Originator shall request to create the resource that have the same content in all members resources belonging to an existing *<group>* resource by using a CREATE operation. The Request may address the virtual child resource *<fanOutPoint>* of the specific *<group>* resource of a group Hosting CSE. The request may also address the address that results from appending a relative address to the *<fanOutPoint>* address in order to create the resources that have the same content under the corresponding child resources represented by the relative address with respect to all members resources. The Originator may be an AE or CSE. |
| Processing at Group Hosting CSE | For the CREATE procedure, the Group Hosting CSE shall:* Check if the Originator has CREATE privilege in the *<accessControlPolicy>* resource referenced by the *membersAccessControlPolicyIDs* in the *<group>* resource. In the case members *membersAccessControlPolicyIDs* is not provided the access control policy defined for the *<group>* resource shall be used
* Upon successful validation, obtain the IDs of all members resources from the attribute *membersIDs* of the addressed *<group>* resource
* If the request contains *groupRequestTargetMembers* parameter, it shall check whether members contains in this parameter are subset of *memberIDs* of the addressed <group> resource. If true, request shall be fanned out to members containing in this parameter only.
* Generate fan out requests addressing the obtained address (appended with the relative address if any) to the member hosting CSEs as indicated in figure 10.2.7.6-1.The ***From*** parameter in the fanout request is set to ID of the Originator from the request from the original Originator. The ***Response Type*** parameter in the fanout request may be set by the group hosting CSE differently according to its local policy
* In the case that a member resource is a *<group>* resource and the request to be fanned out does not contain a group request identifier already, generate a unique group request identifier, include the group request identifier in all the requests to be fanned out and locally store the group request identifier
* If the group Hosting CSE determines that multiple members resources belong to one CSE according to the IDs of the members resources, it may converge the requests accordingly before sending out. This may be accomplished by the group Hosting CSE creating a *<group>* resource on the members Hosting CSE to collect all the members on that members Hosting CSE
* After receiving the responses from the members hosting CSEs, respond to the Originator with the aggregated results and the associated members list. Depending on the ***Response Type***, the Group Hosting CSE shall:

- **blockingRequest:** respond with the aggregated responses before the ***Result Expiration Time*** reaches and discard the member responses received after- **nonBlockingRequestSynch:** prepare the *operationResult* of the <request> resource and indicate that if all the member responses have been aggregated by setting the *requestStatus* of the <request> resource before the ***Result Expiration Time*** reaches. There may be multiple updates of the *operationResult* attribute. - **nonBlockingRequestAsynch:** notify with the aggregated response from all or part of the members before the ***Result Expiration Time*** reaches. There may be more than one notifications.- **flexBlocking:** continue aggregate the member response until the group hosting CSE determines to send the aggregated responses, if all member responses has been aggregated, respond the aggregated response as in the blockingRequest case. Otherwise, respond an acknowledgement together with the current aggregated member responses and the reference to the created <request> resource. Then continue aggregate and deliver the remaining member response to the Originator as defined in the nonBlockingRequestSynch or the nonBlockingRequestAsynch case.- After the ***Result Expiration Time***, there shall not be any further updates to the aggregated responses.(See note) |
| Processing at Member Hosting CSE | For the CREATE procedure, the Member Hosting CSE shall:* Check if the request has a group request identifier. Check if the group request identifier is contained in the requested identifiers stored locally. If match is found, ignore the current request and respond an error. If no match is found, locally store the group request identifier until the expiration of the request expiration time or local policy
* Check if the original Originator has the CREATE permission on the addressed resource. Upon successful validation, perform the create procedures for the corresponding type of addressed resource as described in other sub-clauses of clause 10.2
* Send the corresponding response to the Group Hosting CSE
 |
| Information in Response message | Converged responses from members hosting CSEs |
| Processing at Originator after receiving Response | None |
| Exceptions | * Same request with identical group request identifier received
* Originator does not have the CREATE permission to access the *<fanOutPoint>* resource
* Members in *groupRequestTargetMembers* request parameter are not subset of *memberIDs* of the addressed <group> resource
 |
| NOTE: If ***Result Expiration Time*** is not provide in the original request from the Originator, the group hosting CSE may decide the timer based on its local policy. |

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

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

#### 10.2.7.8 Retrieve *<fanOutPoint>*

This procedure shall be used for retrieving the content of all member resources belonging to an existing *<group>* resource.

Table 10.2.7.8-1: *<fanOutPoint>* RETRIEVE

|  |
| --- |
| *<fanOutPoint>* RETRIEVE  |
| Associated Reference Point | Mca, Mcc and Mcc' |
| Information in Request message | ***From:*** Identifier of the AE or the CSE that initiates the Request***To:*** The address of the *<fanOutPoint>* virtual resource***Content:*** The representation of the resource the Originator intends to retrieve***Group Request Identifier:*** The group request identifier***Response Type:*** If the parameter is set to BlockingSynch, it indicates that the group hosting CSE shall return the aggregated response once. Otherwise if the parameter is set to nonBlockingRequestSynch or nonBlockingRequestAsynch, it indicates that the Group Hosting CSE shall return the aggregated response in a batched mode.***Result Expiration Time:*** Indicates the maximum time limit in which the Group Hosting CSE has to respond the aggregated response. |
| Processing at Originator before sending Request | The Originator shall request to obtain the resource or specific attributes of all member resources belonging to an existing *<group>* resource by using a RETRIEVE operation. The request may address the virtual child resource *<fanOutPoint>* of the specific *<group>* resource of a group Hosting CSE. The request may also address the address that results from appending a relative address to the *<fanOutPoint>* address in order to retrieve the corresponding attributes or child resources represented by the relative address with respect to all members resources. The Originator may be an AE or CSE |
| Processing at Group Hosting CSE | For the RETRIEVE procedure, the Group Hosting CSE shall:* Check if the Originator has RETRIEVE permission in the *<accessControlPolicy>* resource referenced by the *membersAccessControlPolicyIDs* in the addressed *<group>* resource. In the case *membersAccessControlPolicyIDs* is not provided, the access control policy defined for the group resource shall be used
* Upon successful validation, obtain the IDs of all members resources from the *membersIDs* attribute of the addressed *<group>* resource
* If the request contains *groupRequestTargetMembers* parameter, it shall check whether members contains in this parameter are subset of *memberIDs* of the addressed <group> resource. If true, request shall be fanned out to members containing in this parameter only.
* Generate fan out requests addressing the obtained address (appended with the relative address if any) to the members hosting CSEs as indicated in figure 10.2.7.6-1.The ***From*** parameter in the fanout request is set to ID of the Originator from the request from the original Originator. The ***Response Type*** parameter in the fanout request may be set by the group hosting CSE differently according to its local policy
* In the case that a member resource is a *<group>* resource, generate a unique group request identifier and the request to be fanned out does not contain a group request identifier already, include the group request identifier in all the requests to be fanned out and locally store the group request identifier
* If the group hosting CSE determines that multiple members resources belong to one CSE according to the IDs of the members resources, it may converge the requests accordingly before sending out. This may be accomplished by the group Hosting CSE creating a *<group>* resource on the members Hosting CSE to collect all the members on that members Hosting CSE
* After receiving the responses from the members hosting CSEs, respond to the Originator with the aggregated results and the associated members list. Depending on the ***Response Type***, the Group Hosting CSE shall:

- BlockingRequest: respond with the aggregated responses before the ***Result Expiration Time*** reaches and discard the member responses received after.- nonBlockingRequestSynch: prepare the *operationResult* of the <request> resource and indicate that if all the member responses have been aggregated by setting the *requestStatus* of the <request> resource before the ***Result Expiration Time*** reaches. There may be multiple updates of the *operationResult* attribute. - nonBlockingRequestAsynch: notify with the aggregated response from all or part of the members before the ***Result Expiration Time*** reaches. There may be more than one notifications. - flexBlocking: continue aggregate the member response until the group hosting CSE determines to send the aggregated responses. If all member responses has been aggregated, respond the aggregated response as in the blockingRequest case. Otherwise, respond an acknowledgement together with the current aggregated member responses and the reference to the created <request> resource. Then continue aggregate and deliver the remaining member response to the Originator as defined in the nonBlockingRequestSynch or the nonBlockingRequestAsynch case.- After the ***Result Expiration Time***, there shall not be any further updates to the aggregated responses.(See note) |
| Processing at Member Hosting CSE | For the RETRIEVE procedure, the Member Hosting CSE shall:* Check if the request has a group request identifier. Check if the group request identifier is contained in the requested identifier stored locally. If match is found, ignore the current request and respond an error. If no match is found, locally store the request identifier until the expiration of the request expiration time or local policy
* Check if the original Originator has the RETRIEVE permission on the addressed resource. Upon successful validation, perform the retrieve procedures for the corresponding type of addressed resource as described in other sub-clauses of clause 10.2
* Send the corresponding response to the group Hosting CSE
 |
| Information in Response message | Converged responses from members hosting CSEs |
| Processing at Originator after receiving Response | None |
| Exceptions | * Same request with identical group request identifier received
* Originator does not have RETRIEVE permission to access the *<fanOutPoint>* resource
* Members in *groupRequestTargetMembers* request parameter are not subset of *memberIDs* of the addressed <group> resource
 |
| NOTE: If ***Result Expiration Time*** is not provide in the original request from the Originator, the group hosting CSE may decide the timer based on its local policy. |

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

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

#### 10.2.7.9 Update *<fanOutPoint>*

This procedure shall be used for updating the content of all member resources belonging to an existing *<group>* resource.

Table 10.2.7.9-1: *<fanOutPoint>* UPDATE

| *<fanOutPoint>* UPDATE  |
| --- |
| Associated Reference Point | Mca, Mcc and Mcc' |
| Information in Request message | ***From:*** Identifier of the AE or the CSE that initiates the Request***To:*** The address of the *<group>* resource***Content:*** The representation of the resource the Originator intend to Update***Group Request Identifier:*** The group request identifier***Response Type:*** If the parameter is set to BlockingSynch, it indicates that the group hosting CSE shall return the aggregated response once. Otherwise if the parameter is set to nonBlockingRequestSynch or nonBlockingRequestAsynch, it indicates that the Group Hosting CSE shall return the aggregated response in a batched mode***Result Expiration Time:*** Indicates the maximum time limit in which the Group Hosting CSE has to respond the aggregated response. |
| Processing at Originator before sending Request | The Originator shall request to update all member resources belonging to an existing *<group>* resource with the same data by using a UPDATE operation. The request may address the virtual child resource *<fanOutPoint>* of the specific *<group>* resource of a group Hosting CSE. The request may also address the address that results from appending a relative address to the *<fanOutPoint>* in order to update the corresponding child resources represented by the relative address with respect to all *<members>* resources. The Originator may be an AE or CSE. |
| Processing at Group Hosting CSE | For the UPDATE procedure, the Group Hosting CSE shall:* Check if the Originator has UPDATE permission in the *<accessControlPolicy>* resource referenced by the *membersAccessControlPolicyIDs* in the group resource. In the case members *membersAccessControlPolicyIDs* is not provided the access control policy defined for the group resource shall be used
* Upon successful validation, obtain the IDs of all member resources from the attribute *membersIDs* of the addressed *<group>* resource
* If the request contains *groupRequestTargetMembers* parameter, it shall check whether members contains in this parameter are subset of *memberIDs* of the addressed <group> resource. If true, request shall be fanned out to members containing in this parameter only.
* Generate fan out requests addressing the obtained address (appended with the relative address if any) to the members hosting CSEs as indicated in figure10.2.7.6-1.The ***From*** parameter in the fanout request is set to ID of the Originator from the request from the original Originator. The ***Response Type*** parameter in the fanout request may be set by the group hosting CSE differently according to its local policy
* In the case that a member resource is a *<group>* resource and the request to be fanned out does not contain a group request identifier already, generate a unique group request identifier, include it in all the requests to be fanned out and locally store the group request identifier
* If the group Hosting CSE determines that multiple members resources belong to one CSE according to the IDs of the member resources, it may converge the requests accordingly before sending out. This may be accomplished by the group Hosting CSE creating a *<group>* resource on the member Hosting CSE to collect all the members on that members Hosting CSE
* After receiving the responses from the members hosting CSEs, respond to the Originator with the aggregated results and the associated members list. Depending on the ***Response Type***, the Group Hosting CSE shall:

- BlockingRequest: respond with the aggregated responses before the ***Result Expiration Time*** reaches and discard the member responses received after- nonBlockingRequestSynch: prepare the *operationResult* of the <request> resource and indicate that if all the member responses have been aggregated by setting the *requestStatus* of the <request> resource before the ***Result Expiration Time*** reaches. There may be multiple updates of the *operationResult* attribute. - nonBlockingRequestAsynch: notify with the aggregated response from all or part of the members before the ***Result Expiration Time*** reaches. There may be more than one notifications. - flexBlocking: continue aggregate the member response until the group hosting CSE determines to send the aggregated responses, if all member responses has been aggregated, respond the aggregated response as in the blockingRequest case. Otherwise, respond an acknowledgement together with the current aggregated member responses and the reference to the created <request> resource. Then continue aggregate and deliver the remaining member response to the Originator defined in the nonBlockingRequestSynch or the nonBlockingRequestAsynch case- After the ***Result Expiration Time***, there shall not be any further updates to the aggregated responses.(See note) |
| Processing at Member Hosting CSE | For the UPDATE procedure, the Member Hosting CSE shall:* Check if the request has a group request identifier. Check if the request identifier is contained in the requested identifier stored locally. If match is found, ignore the current request and respond an error. If no match is found, locally store the request identifier until the expiration of the request expiration time or local policy
* Check if the original Originator has the UPDATE permission on the addressed resource. Upon successful validation, perform the update procedures for the corresponding type of addressed resource as described in other sub-clauses of clause 10.2
* Send the corresponding response to the group Hosting CSE
 |
| Information in Response message | Converged responses from members hosting CSEs |
| Processing at Originator after receiving Response | None |
| Exceptions | * Same request with identical group request identifier received
* Originator does not have the UPDATE permissions to access the *<fanOutPoint>* resource
* Members in *groupRequestTargetMembers* request parameter are not subset of *memberIDs* of the addressed <group> resource
 |
| NOTE: If ***Result Expiration Time*** is not provide in the original request from the Originator, the group hosting CSE may decide the timer based on its local policy. |

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

### -----------------------Start of change 4----------------------------------------------

#### 10.2.7.10 Delete *<fanOutPoint>*

This procedure shall be used for deleting the content of all members resources belonging to an existing *<group>* resource.

Table 10.2.7.10-1: *<fanOutPoint>* DELETE

| *<fanOutPoint>* DELETE  |
| --- |
| Associated Reference Point | Mca, Mcc and Mcc' |
| Information in Request message | ***From:*** Identifier of the AE or the CSE that initiates the Request***To:*** The address of the *<fanOutPoint>* virtual resource***Content:*** The representation of the resource the Originator intends to delete***Group Request Identifier:*** The group request identifier***Response Type:*** If the parameter is set to BlockingSynch, it indicates that the group hosting CSE shall return the aggregated response once. Otherwise if the parameter is set to nonBlockingRequestSynch or nonBlockingRequestAsynch, it indicates that the Group Hosting CSE shall return the aggregated response in a batched mode***Result Expiration Time:*** Indicates the maximum time limit in which the Group Hosting CSE has to respond the aggregated response. |
| Processing at Originator before sending Request | The Originator shall request to delete all members resources belonging to an existing *<gro*u*p>* resource by using a DELETE operation. The request may address the virtual child resource *<fanOutPoint>* of the specific *<group>* resource of a group Hosting CSE. The request may also address the address that results from appending a relative address to the *<fanOutPoint>* in order to delete the corresponding child resources represented by the relative address with respect to all member resources. The Originator may be an AE or a CSE |
| Processing at Group Hosting CSE | For the DELETE procedure, the *<group>* Hosting CSE shall:* Check if the Originator has DELETE permission in the *<accessControlPolicy>* resource referenced by the *membersAccessControlPoliciIDs* in the *<group>* resource. In the case *membersAccessControlPolicyIDs* is not provided the access control policy defined for the group resource shall be used
* Upon successful validation, obtain the IDs of all member resources from the attribute *membersIDs* of the addressed *<group>* resource
* If the request contains *groupRequestTargetMembers* parameter, it shall check whether members contains in this parameter are subset of *memberIDs* of the addressed <group> resource. If true, request shall be fanned out to members containing in this parameter only.
* Generate fan out requests addressing the obtained address (appended with the relative address if any) to the member hosting CSEs as indicated in figure 10.2.7.6-1. ***From*** parameter in the fanout request is set to ID of the Originator from the request from the original Originator. The ***Response Type*** parameter in the fanout request may be set by the group hosting CSE differently according to its local policy
* In the case that the members resources is a *<group>* resource and the request to be fanned out does not contain a group request identifier already, generate a unique group request identifier, include the group request identifier in all the requests to be fanned out and locally store the group request identifier
* If the *<group>* Hosting CSE determines that multiple members resources belong to one CSE according to the IDs of the members resources, it may converge the requests accordingly before sending out. This may be accomplished by the group Hosting CSE creating a *<group>* resource on the member Hosting CSE to collect all the members on that member Hosting CSE
* After receiving the responses from the members hosting CSEs, respond to the Originator with the aggregated results and the associated members list. Depending on the ***Response Type***, the Group Hosting CSE shall:

- BlockingRequest: respond with the aggregated responses before the ***Result Expiration Time*** reaches and discard the member responses received after- nonBlockingRequestSynch: prepare the *operationResult* of the <request> resource and indicate that if all the member responses have been aggregated by setting the *requestStatus* of the <request> resource before the ***Result Expiration Time*** reaches. There may be multiple updates of the *operationResult* attribute.- nonBlockingRequestAsynch: notify with the aggregated response from all or part of the members before the ***Result Expiration Time*** reaches. There may be more than one notifications.- flexBlocking: continue aggregate the member response until the group hosting CSE determines to send the aggregated responses, if all member responses has been aggregated, respond the aggregated response as in the blockingRequest case. Otherwise, respond an acknowledgement together with the current aggregated member responses and the reference to the created <request> resource. Then continue aggregate and deliver the remaining member response to the Originator as defined in the nonBlockingRequestSynch or the nonBlockingRequestAsynch case- After the ***Result Expiration Time***, there shall not be any further updates to the aggregated responses(See note) |
| Processing at Member Hosting CSE | For the DELETE procedure, the Members Hosting CSE shall:* Check if the request has a group request identifier. Check if the group request identifier is contained in the requested identifier stored locally. If match is found, ignore the current request and respond an error. If no match is found, locally store the group request identifier until the expiration of the request expiration time or local policy
* Check if the original Originator has the DELETE permission on the addressed resource. Upon successful validation, perform the delete procedures for the corresponding type of addressed resource as described in other sub-clauses of clause 10.2
* Send the corresponding response to the Group Hosting CSE
 |
| Information in Response message | Converged responses from members hosting CSEs |
| Processing at Originator after receiving Response | None |
| Exceptions | * Same request with identical group request identifier received
* Originator does not have the DELETE permissions to access the *<fanOutPoint>* resource
* Members in *groupRequestTargetMembers* request parameter are not subset of *memberIDs* of the addressed <group> resource
 |
| NOTE: If ***Result Expiration Time*** is not provide in the original request from the Originator, the group hosting CSE may decide the timer based on its local policy. |

### -----------------------End of change 4----------------------------------------------

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 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.?