|  |
| --- |
|  |

|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* | SDS 46 |
| Source:\* | Marianne Mohali, Orange, [Marianne.mohali@orange.com](mailto:Marianne.mohali@orange.com) |
| Date:\* | 2020-07-07 |
| Reason for Change/s:\* | Discovery-based operations |
| 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)  STE Small Technical Enhancements / < Work Item number (optional)>  Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0004 v4.1.0 |
| Clauses \* | 6.3.4.2.7, 6.3.4.2.31, 6.6.3.5, 7.2.1.1, 7.2.1.2, 7.2.2.2, 7.3.3.1, 7.3.3.14, 7.2.2.17, 7.2.2.17.0(new), 7.5.2. |
| 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 | <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 2020 (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.

If this is a correction, and the change applies to previous releases, a separate “mirror CR” should be posted at the same time as 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. Include any changes to references, definitions, and abbreviations in the same deliverable.

Follow the drafting rules.

All pictures must be editable.

Check spelling and grammar.

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 proposed new clause is 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 the 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 was intended to be a companion CR to ARC-2018-0348, implementing protocol -level details for Discovery-based operations, as well as some corrections and clarifications for the discovery operation. CR against TS-0001 were agreed but the CR for TS-0004 is missing.

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

##### 6.3.4.2.7 m2m:resultContent

Used for ***Result Content*** parameter in the request primitive.

Table 6.3.4.2.7‑1: Interpretation of resultContent

|  |  |  |
| --- | --- | --- |
| **Value** | **Interpretation** | **Note** |
| 0 | nothing |  |
| 1 | attributes |  |
| 2 | hierarchical address |  |
| 3 | hierarchical address and attributes |  |
| 4 | attributes and child resources |  |
| 5 | attributes and child resource references |  |
| 6 | child resource references |  |
| 7 | original resource |  |
| 8 | child resources |  |
| 9 | modified attributes |  |
| 10 | semantic content |  |
| 11 | discovery result resource references |  |
| NOTE: See clause 6.4.1 Request primitive parameter data types. | | |

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

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

##### 6.3.4.2.31 m2m:filterUsage

Used in m2m:filterCriteria.

Table 6.3.4.2.31‑1: Interpretation of filterUsage

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | Discovery |  |
| 2 | Conditional Operation | This is the default value when the filterUsage condition is not present in a request. |
| 3 | IPE On-demand Discovery |  |
| 4 | Discovery-based Operation |  |

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

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

#### 6.6.3.5 Originator error response class

Table 6.6.3.5-1 specifies the RSCs for Originator error responses.

41xx codes are oneM2M specific.

Table 6.6.3.5‑1: RSCs for Originator error response class

|  |  |
| --- | --- |
| Numeric Code | Description |
| 4000 | BAD\_REQUEST |
| 4001 | RELEASE\_VERSION\_NOT\_SUPPORTED |
| 4004 | NOT\_FOUND |
| 4005 | OPERATION\_NOT\_ALLOWED |
| 4008 | REQUEST\_TIMEOUT |
| 4015 | UNSUPPORTED\_MEDIA\_TYPE |
| 4101 | SUBSCRIPTION\_CREATOR\_HAS\_NO\_PRIVILEGE |
| 4102 | CONTENTS\_UNACCEPTABLE |
| 4103 | ORIGINATOR\_HAS\_NO\_PRIVILEGE |
| 4104 | GROUP\_REQUEST\_IDENTIFIER\_EXISTS |
| 4105 | CONFLICT |
| 4106 | ORIGINATOR\_HAS\_NOT\_REGISTERED |
| 4107 | SECURITY\_ASSOCIATION\_REQUIRED |
| 4108 | INVALID\_CHILD\_RESOURCE\_TYPE |
| 4109 | NO\_MEMBERS |
| 4110 | GROUP\_MEMBER\_TYPE\_INCONSISTENT |
| 4111 | ESPRIM\_UNSUPPORTED\_OPTION |
| 4112 | ESPRIM\_UNKNOWN\_KEY\_ID |
| 4113 | ESPRIM\_UNKNOWN\_ORIG\_RAND\_ID |
| 4114 | ESPRIM\_UNKNOWN\_RECV\_RAND\_ID |
| 4115 | ESPRIM\_BAD\_MAC |
| 4116 | ESPRIM\_IMPERSONATION\_ERROR |
| 4117 | ORIGINATOR\_HAS\_ALREADY\_REGISTERED |
| 4118 | ONTOLOGY\_NOT\_AVAILABLE |
| 4119 | LINKED\_SEMANTICS\_NOT\_AVAILABLE |
| 4120 | INVALID\_SEMANTICS |
| 4121 | MASHUP\_MEMBER\_NOT\_FOUND |
| 4122 | INVALID\_TRIGGER\_PURPOSE |
| 4123 | ILLEGAL\_TRANSACTION\_STATE\_TRANSITION\_ATTEMPTED |
| 4124 | BLOCKING\_SUBSCRIPTION\_ALREADY\_EXISTS |
| 4125 | SPECIALIZATION\_SCHEMA\_NOT\_FOUND |
| 4126 | APP\_RULE\_VALIDATION\_FAILED |
| 4127 | OPERATION\_DENIED\_BY\_REMOTE\_ENTITY |
| 4130 | ONTOLOGY\_MAPPING\_ALGORITHM\_NOT\_AVAILABLE |
| 4131 | ONTOLOGY\_MAPPING\_POLICY\_NOT\_MATCH |
| 4132 | ONTOLOGY\_MAPPING\_NOT\_AVAILABLE |
| 4133 | BAD\_FACT\_INPUTS\_FOR\_REASONING |
| 4134 | BAD\_RULE\_INPUTS\_FOR\_REASONING |
| 4135 | DISCOVERY\_LIMIT\_EXCEEDED |

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

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

#### 7.2.1.1 Request primitive format

Table 7.2.1.1‑1 summarizes the primitive parameters of the Request primitive, indicating their presence depending on the C, R, U, D or N operations. "M" indicates mandatory, "O" indicates optional, "NP" indicates not present.

Refer to clause 8.1.2 of the oneM2M TS-0001 [6] for additional information on the request primitive parameters.

Table 7.2.1.1‑1: Request Primitive Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Primitive Parameter** | **CREATE** | **RETRIEVE** | **UPDATE** | **DELETE** | **NOTIFY** |
| Operation | M | M | M | M | M |
| To | M | M | M | M | M |
| From | O  See note | M | M | M | M |
| Request Identifier | M | M | M | M | M |
| Resource Type | M | NP | NP | NP | NP |
| Content | M | O | M | NP | M |
| Role IDs | O | O | O | O | O |
| Originating Timestamp | O | O | O | O | O |
| Request Expiration Timestamp | O | O | O | O | O |
| Result Expiration Time | O | O | O | O | O |
| Operation Execution Time | O | O | O | O | O |
| Response Type | O | O | O | O | O |
| Result Persistence | O | O | O | O | NP |
| Result Content | O | O | O | O | NP |
| Event Category | O | O | O | O | O |
| Delivery Aggregation | O | O | O | O | O |
| Group Request Identifier | O | O | O | O | O |
| Filter Criteria | NP | O | O | O | NP |
| Desired Identifier Result Type | O | O | O | O | NP |
| Tokens | O | O | O | O | O |
| Token IDs | O | O | O | O | O |
| Local Token IDs | O | O | O | O | O |
| Token Request Indicator | O | O | O | O | O |
| Group Request Target Members | O | O | O | O | NP |
| Authorization Signature Indicator | O | O | O | O | NP |
| Authorization Signature | O | O | O | O | NP |
| Authorization Relationship Indicator | O | O | O | O | NP |
| Semantic Query Indicator | NP | O | NP | NP | NP |
| Release Version Indicator | M | M | M | M | M |
| Vendor Information | O | O | O | O | O |
| NOTE: The *From* parameter is Mandatory for all requests except for AE CREATE. For AE CREATE, it is Optional. | | | | | |

The ***Content*** parameter in a Request shall contain one of the following:

1. A partial Resource. This applies to Create and Update request primitives. In the case of Create request the ***Content*** parameter shall contain a single root element whose name is the name of the Resource and whose content consists of one or more attributes, child Resources or childResource references. In the case of an Update request primitive, the ***Content*** parameter shall contain the attribute and new values. Attributes to be deleted from the resource shall be indicated without a value. In both cases the resource type is as defined in clause 7.4, however since a partial resource is being transferred it is not required to be valid according to the XSD for that resource in terms of the presence of resource attributes. Any attribute that is present, however, shall comply to the data type defined in the XSD of that resource.
2. A Notification Data Object. This applies to Notification request primitives. The data type of the data object is named <m2m:notification> and is described in clause 7.5.1.
3. An Aggregated Notification. This applies to Notification request primitives. The data type of the data object is named <m2m:aggregatedNotification> and contains multiple <m2m:notification> objects. This is described in clause 7.5.1.
4. An AttributeList element, as described in clause 7.5.2. This is used in partial retrieve request primitives to indicate a list of attribute names whose values shall be retrieved in the response.
5. A ResponsePrimitive object as described in clause 7.5.1. This applies to Notification request primitives which are sent when accessing resources in asynchronous non-blocking mode.

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

### -----------------------Start of change 5-------------------------------------------

#### 7.2.1.2 Response primitive format

Table 7.2.1.2‑1 summarizes the primitive parameters for Response primitive, indicating their presence depending on the C, R, U, D or N operations of the associated Request primitive and whether this operation was successful or caused an error. "M" indicates mandatory, "O" indicates optional, "NP" indicates not present.

Refer to clause 8.1.3 of oneM2M TS-0001 [6] for additional information on the request primitive parameters.

NOTE: ***Response Code*** and ***Status Code*** parameters are merged into the ***Response Status Code*** parameter.

Table 7.2.1.2‑1: Response Primitive Parameters

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Primitive parameter** | **Ack** | **CREATE**  **Success** | **RETRIEVE**  **Success** | **UPDATE**  **Success** | DELETE  Success | NOTIFY  Success | Error |
| Response Status Code | M | M | M | M | M | M | M |
| Request Identifier | M | M | M | M | M | M | M |
| Content | O | O | M | O | O | O | O |
| To | O | O | O | O | O | O | O |
| From | O | O | O | O | O | O | O |
| Originating Timestamp | O | O | O | O | O | O | O |
| Result Expiration Timestamp | O | O | O | O | O | O | O |
| Event Category | O | O | O | O | O | O | O |
| Content Status | NP | NP | O | NP | NP | NP | NP |
| Content Offset | NP | NP | O | NP | NP | NP | NP |
| Assigned Token Identifiers | NP | O | O | O | O | O | O |
| Token Request Information | NP | NP | NP | NP | NP | NP | O |
| Authorization Signature Request Information | NP | NP | NP | NP | NP | NP | O |
| Release Version Indicator | M | M | M | M | M | M | M |
| Vendor Information | O | O | O | O | O | O | O |

The Content parameter in a Response shall contain one of the following:

1. A complete or partial Resource. This applies to a response primitive sent in reply to create and retrieve request message. A partial resource also applies to a response primitive sent in reply to update request message. The ***Content*** parameter shall contain a single root element whose name is the name of the Resource and whose content consists of one or more attributes, child resources or childResource references. In this case the resource type is as defined in clause 7.4. However if a partial resource is being transferred, it is not required to be valid according to the XSD for that resource, in terms of the presence of resource attributes. Any attribute that is present, however, shall comply to the data type defined in the XSD of that resource.
2. The URI of a resource. This is included directly as the content of the ***Content*** parameter (like in case 6).
3. A partial resource and its hierarchical URI. These are included in a root element called m2m:resource defined in clause 7. The URI is included as an attribute of m2m:resource.
4. A list of URIs. This can be used for transferring the childResource URIs in a Discovery response. These are included in an element called m2m:URIList defined in clause 7.
5. A list of childResourceRef. This can be used for transferring the child resource references in a Discovery response. These are included in an element called m2m:resourceRefList defined in clause 7.
6. An Aggregated Response. This is sent as a result of a Group operation or as a response to a Discovery-based Operation with *Result Content* not present or different than "discovery Result Resource References". The response uses the element m2m:aggregatedResponse defined in clause 7.
7. A request primitive. A pending request is sent in a polling response. This uses the element m2m:requestPrimitive defined in clause 6.4.1.
8. Human-readable error message. This is included in an element called m2m:debugInfo defined in clause 7.

### -----------------------End of change 5---------------------------------------------

### -----------------------Start of change 6-------------------------------------------

#### 7.2.2.2 Generic procedure for handling a Request at a receiver

The Receiver shall execute the following steps in order. In case of error in any of the steps below, the Receiver shall execute "Create an error response" (refer to clause 7.3.3.13 for details) and then "Send Response primitive" (refer to clause 7.3.2.4 for details). The corresponding ***Response Status Code*** shall be included in the Response primitive.

Recv-1.0: "Check the validity of received request primitive"

Recv-2.0: Communication method?

Recv-6.0: Resource handling procedures

Recv-3.0: "Create <request> resource locally"

Recv-4.0: "Create a success Response"

nonBlockingRequestAsynch

nonBlockingRequestSynch

Recv-6.0: Resource handling procedures

Recv-7.0: "Update <request> resource"

Finish

Start

Recv-3.0: "Create <request> resource locally"

Recv-4.0: "Create a success Response"

Recv-6.0: Resource handling procedures

Recv-7.0: "Update <request> resource"

Recv-8.0: "Send Notification"

Recv-9.0: "Wait for Response primitive"

blockingRequest

Recv-10.0: "Send Response Primitive"

Recv-5.0: "Send Response primitive"

Recv-5.0: "Send Response primitive"

Figure 7.2.2.2‑1: Generic procedure of Receiver

Recv-1.0 "Check the validity of received request primitive": See clause 7.3.2.1 for details.

Recv-2.0 "Communication method?": The Receiver CSE checks whether a received request is blockingRequest, nonBlockingRequestSynch or nonBlockingRequestAsynch by using the ***Response Type*** parameter (see detail in clause 8.1.2 in oneM2M TS-0001 [6]). If the request is blockingRequest or the ***Response Type*** parameter is not included, it goes to step Recv-6.0 "Resource handling procedure". If the request is nonBlockingRequestSynch, it goes to step Recv‑3.0 "Create <request> resource locally". If the request is nonBlockingRequestAsynch, it goes to step Recv‑3.0 "Create <request> resource locally". If the request is flexBlocking, the Receiver CSE shall make the decision to respond using blocking or non-blocking based on its own local context (memory, processing capability, etc.) unless specified further in the resource-specific procedure.

Recv-3.0 "Create <request> resource locally": Refer to clause 7.3.2.2 for details.

Recv-4.0 "Create a successResponse": Refer to clause 7.3.3.12 for details.

Recv-5.0 "Send Response Primitive": Refer to clause 7.3.2.4 for details.

Recv-6.0 "Resource handling procedure": Refer to Figure 7.2.2.2‑2 for details.

Recv-7.0 "Update <request> resource": Refer to clause 7.3.2.5 for details. This step is only valid when the request is non-blocking.

Recv-8.0 "Send Notification": Refer to clause 7.5.1.2.5 for details.

Recv-9.0 "Wait for a Response primitive": Refer to clause 7.3.1.3 for details.

Recv-10.0 "Send Response Primitive": Refer to clause 7.3.3.16 for details.

Recv-6.10: "Queue request primitive and execute CMDH message forwarding procedure"

Recv-6.1: Hosting CSE of the targeted resource?

Start

Recv-6.3: "Check authorization of the Originator"

Recv-6.2: "Check existence of the addressed resource"

Finish

Recv-6.4: "Check validity of resource representation for the given resource type"

Recv-6.5: "Create/Update/Retrieve/Delete/Notify operation is performed"

Yes

No

Recv-6.6: "Announce/De-announce the resource"

Recv-6.9: CMDH processing supported?

Recv-6.11: "Forwarding"

No

Yes

Recv-6.0.1: Requested operation is an AE registration?

Recv-6.0.2: "Check Service Subscription Profile"

Yes

No

Else

blockingRequest

Recv-6.6.1: "Communication Method?"

Recv-6.7: "Create a success response"

Recv-6.2.1: "Check for duplicate group requests"

Recv-6.10: "Queue request primitive and execute CMDH message forwarding procedure"

Recv-6.1: Hosting CSE of the targeted resource?

Start

Recv-6.3: "Check authorization of the Originator"

Recv-6.4: "Check validity of resource representation for the given resource type"

Recv-6.2: "Check existence of the targeted resource"

Recv-6.5: "Create/Update/Retrieve/Delete/Notify operation is performed"

Recv-6.6: "Announce/De-announce the resource"

Finish

Yes

No

Recv-6.7: "Create a success response"

Recv-6.9: CMDH processing supported?

Recv-6.11: "Forwarding"

No

Yes

Recv-6.0.1: Requested operation is an AE registration?

Recv-6.0.2: "Check Service Subscription Profile"

Yes

No

Recv-6.6.1: "Communication Method?"

Else

blockingRequest

Figure 7.2.2.2‑2: Resource handling procedure

Figure 7.2.2.2-2 describes the generic procedure to resource handling procedures.

Recv-6.0.1 "Requested operation is an AE registration?": If the requested operation is an AE registration, then it goes to Recv-6.0.2 "Check Service Subscription Profile". Otherwise, it goes to Recv-6.1.

Recv-6.0.2 "Check Service Subscription Profile": Refer to clause 7.3.2.7 for details.

Recv-6.1 "Hosting CSE of the targeted resource?": The step checks if the receiver is a transit CSE or the Hosting CSE of the received Request by examining the ***To*** parameter of the Request primitive. If the receiver hosts the resource that the address in the ***To*** parameter represents, the receiver is the Hosting CSE (goes to Recv-6.2"Check existence of the addressed resource", Yes branch). Otherwise, the receiver is the Transit CSE (goes to Recv-6.9 "CMDH processing supported?", No branch). Refer to clause 7.3.2.8 for details.

Recv-6.1.1 "*filterUsage?*": The Hosting CSE checks if the request includes steps for discovery or identification of target resources or not by examining the ***filterUsage*** handling criterion.

If ***filterUsage*** is configured as “Conditional Operation" or is not present, it goes to Recv-6.2 "Check existence of targeted resource".

If ***filterUsage*** is configured as "Discovery" or "IPE On-demand Discovery" it goes to Recv-6.1.2 "Discovery-related procedure". Refer to clause 7.3.3.14 for details.

If ***filterUsage*** is configured as "Discovery-based Operation" it goes to Recv-6.1.2 "Discovery-related procedure" where more than one targeted resources may be identified and the following processing repeats for each identified target resource.

Recv-6.1.2 "Discovery-related procedure": Refer to clause 7.3.3.14 for details

Recv-6.2 "Check existence of the targeted resource": Refer to clause 7.3.3.2 for details.

Recv-6.3 "Check authorization of the Originator": Refer to clause 7.3.3.15 for details.

Recv-6.4 "Check validity of resource representation": Refer to clause 7.3.3.3 and clause 7.3.3.4 for details. Notify is not applicable for this step.

Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed": The step represents five common operations which are "Create the resource (clause 7.3.3.5)", "Retrieve the resource (clause 7.3.3.6)", "Update the resource (clause 7.3.3.7)", "Delete the resource (clause 7.3.3.8)" and "Notify processing (clause 7.3.3.9)".

Recv-6.6 "Announce/De-announce the resource": The step represents two common operations which are "Announce the resource" and "De-announce the resource". Refer to clause 7.3.3.10 and clause 7.3.3.11 for details. Notify is not applicable for this step.

Recv-6.6.1 "Communication method?": The Receiver CSE checks whether a received request is blockingRequest or not by using ***Response Type*** parameter (see detail in clause 8.1.2 in oneM2M TS-0001 [6]). If the request was blockingRequest or ***Response Type*** parameter was not included, it goes to step Recv-6.7 "Create a success response". Otherwise, it goes back to the generic procedure of the receiver (Figure 7.2.2.2‑1).

Recv-6.7 "Create a success response": Refer to clause 7.3.3.12 for details.

Recv-6.9 "CMDH processing supported?": This step checks whether the Receiver supports the CMDH processing. If the receiver supports CMDH processing, it goes to Recv-6.10 "Queue request primitive and execute CMDH message forwarding procedure" otherwise, it goes to Recv-6.11 "Forwarding".

Recv-6.10 "Queue request primitive and execute CMDH message forwarding procedure": the Receiver CSE shall queue the received request primitive and execute the "CMDH message forwarding procedure". Refer to clause H.2.4 for details.

Recv-6.11 "Forwarding": carry out message forwarding as defined in clause 7.3.2.6.

### -----------------------End of change 6---------------------------------------------

### -----------------------Start of change 7-------------------------------------------

#### 7.3.3.1 Check existence of the targeted resource

If the ***Request Expiration Timestamp*** is given in the request and has expired, the Hosting CSE shall reject the request with a "REQUEST\_TIMEOUT" ***Response Status Code*** parameter value. Otherwise, the Hosting CSE should handle the request before the time specified in ***Request Expiration Timestamp.***

The Hosting CSE shall check if the resource addressed by the ***To*** parameter exists in the repository. If the resource does not exist, the Hosting CSE shall reject the request with a ***Response Status Code*** indicating "NOT\_FOUND" error.

The Hosting CSE shall also check the existence of the target resource based on any ***Filter Criteria*** ***Matching Conditions*** present in the Create/Retrieve/Update/Delete request. If there is no matching target resource, the Hosting CSE shall reject the request with a ***Response Status Code*** indicating "NOT\_FOUND" error.

If the Hosting CSE does not support the content format (i.e. type of serialization) requested by the originator, the request shall be rejected with a ***Response Status Code*** indicating "NOT\_ACCEPTABLE" error.

If the Hosting CSE does not support the content format sent by the originator, the request shall be rejected with a ***Response Status Code*** indicating "UNSUPPORTED\_MEDIA\_TYPE" error.

### -----------------------End of change 7---------------------------------------------

### -----------------------Start of change 8-------------------------------------------

#### 7.3.3.14 Discovery related procedure

If the Operation Execution Time is given in the request, the Hosting CSE should perform the following procedures at the time and shall not perform the procedures before the time.

The Discovery procedure is used to discover resources in a CSE. A Discovery-related request is done by sending a Retrieve request with ***filterUsage***, one of the ***Filter Criteria*** parameters, configured as "Discovery" and the request may include other ***Filter Criteria*** parameters as well. A resource discovery request procedure shall be comprised of the following actions.

If the parameter ***filterUsage*** is configured as "Discovery", "Discovery-based Operation " or "IPE On-demand Discovery" the request is a Discovery-related request and the Hosting CSE shall perform the steps described below. The parameter ***filterUsage*** indicates the type of procedure to be performed, as follows:

* Discovery: initiated by a Retrieve request with ***filterUsage*** configured as "Discovery".
* Discovery-based Operation: initiated by Create, Update or Delete requests with ***filterUsage*** configured as "Discovery-based Operation".

IPE On-demand Discovery: initiated by a Retrieve request with ***filterUsage*** configured as "IPE On-demand Discovery".The Hosting CSE shall perform the following procedures in order to identify the resource set based on conditions specified in ***Filter Criteria***.

The Discovery procedure begins with the children of the resource addressed by the ***To*** parameter (the ‘targeted resource’). The Hosting CSE shall not include the targeted resource in the resulting resource set.

1. The Hosting CSE shall check if the resource addressed by the To parameter exists. If the resource does not exist, the Hosting CSE shall reject the request with a Response Status Code indicating "NOT\_FOUND" error.

The Hosting CSE shall examine the child / descendent resources of the targeted resource to find the resources that match the Filter Criteria Matching Conditions. In this search, the Hosting CSE will not consider any child/descendent <AE> resource with registrationStatus attribute set to INACTIVE, and any child/descendent resources of this INACTIVE <AE> resource. The Hosting CSE shall proceed with the search even if the targeted resource does not match the Matching Conditions.

1. The scope of the search is subject to filter handling conditions if specified by the Originator in the ***Filter Criteria*** as follows (see also clause 8.1.2 of oneM2M TS-0001 [6]):

* Based on the filter handling condition ***level***: The Hosting CSE shall skip over and not include any descendants in the resource tree that are deeper than the maximum level indicated by the ***level*** condition. This level is measured from the targeted resource (i.e. ***To*** parameter).
* Based on the filter handling condition ***offset***: At the start of its search the Hosting CSE shall skip over and not include in the number of child and descendant resources indicated by the ***offset*** condition. The ***offset*** condition is permitted only for Retrieve operations.

1. For ***filterUsage*** “Discovery” or “Discovery-based Operation'” if the filter handling condition ***applyRelativePath***is provided, the Hosting CSE appends the specified relative path to each resource in the matching result. The matching resource is excluded and is replaced by the resource corresponding to this compounded path, if there is one. The replacement resource is not required to match the filter Matching Conditions. See 7.3.3.17.17 for details about ***applyRelativePath****.*
2. The Hosting CSE shall exclude any resources from the resource set to which the Originator does not have "Discover" privilege.
3. If the request is a retrieve request with ***filterUsage*** set to "IPE On-demand Discovery” and addressing an <AE> resource representing the IPE by its *labels* attribute, the Hosting CSE shall first find resources using the ***Filter Criteria*** as detailed above. When the Hosting CSE finds no match, the Hosting CSE shall check the filterUsage element.If the Hosting CSE finds no matches, the Hosting CSE shall send the Notify request to the IPE to trigger the external discovery procedure (see clause 7.5.1.2.8 for more details). If the Hosting CSE receives a successful Notify response, the Hosting CSE shall check the Originator's "Discover" privilege for the resources listed in the Notify response. If the Hosting CSE receives an unsuccessful Notify response from the IPE, then the Hosting CSE shall use the same ***Response Status Code*** in the response to the Originator.
4. The final resource set shall be truncated if it contains more than the maximum permitted number of resources. This maximum shall be the smaller of:
   * The Originator-supplied filter handling condition ***limit***, if there is one
   * A value set by the Hosting CSE based on local policies.

If ***filterUsage*** is configured as “Discovery” or "IPE On-demand Discovery" the Hosting CSE shall proceed to step Recv-6.7 “Create a success response”. The Hosting CSE shall include in the response addresses for all the resources in the resource set produced by steps 1-6 above. It shall use the resource identifier format specified by the ***Desired Identifier Result Type*** parameter setting (see clause 6.3.4.2.8).

* If there are no resources in the resource target set, the Hosting CSE shall respond with successful response with an empty Content parameter as follows:
  + If rcn is not present the ***primitive Content*** is an m2m:URIList containing no entries
  + If rcn=6 the ***primitive Content*** is an m2m:resourceRefList containing no entries
* If the resource set was truncated in step 6 the response shall contain a ***Content Status*** parameter with the value PARTIAL\_CONTENT, and a ***Content Offset*** parameter indicating the point at which the Originator can continue discovery.

If ***filterUsage*** is configured as “Discovery-based Operation" and the resource set was truncated in step 6 by Hosting CSE policy, the Hosting CSE shall reject the request with a ***Response Status Code*** indicating a "DISCOVERY\_LIMIT\_EXCEEDED" error.

If the result set was not truncated, or was truncated to the value of the ***limit*** condition supplied by the Originator, the Hosting CSE shall repeat steps Recv-6.3- Recv-6.6 for each of the resources in the resource set from steps 1-6 as “addressed” or “targeted” resources. After the processing has been repeated for all the resources in the resource set, the Hosting CSE shall proceed to step Recv-6.7 “Create a success response”. The Hosting CSE shall respond with successful response as follows:

* If rcn is not present or other than “discovery result resource references”all the individual primitives created either as successful or as error responses are aggregated to be sent as a single response primitive. If the target resource set is empty the primitive content is an m2m:aggregatedResponse containing no entries in the successful response.
* If rcn=”discovery result resource references” a single response primitive is composed, with content that includes the resource references of all the target resources. If the target resource set is empty the ***primitive content*** is an m2m:URIList containing no entries in the successful response.

### -----------------------End of change 8---------------------------------------------

### -----------------------Start of change 9-------------------------------------------

#### 7.3.3.17 Using Filter Criteria for identification of target resources

##### 7.3.3.17.0 Introduction

When the Filter Criteria primitive parameter is present in a request primitive, it shall be applied for identification of the applicable target resources of the respective operation. This may apply to Create, Retrieve, Update, Delete, Discovery and Semantic Resource Discovery operations as specified in clauses 7.3.3.5, 7.3.3.6, 7.3.3.7, 7.3.3.8, 7.3.3.14 and 7.3.3.18 respectively.

The Filter Criteria primitive parameter defines **Matching Conditions** on resource attributes or content (table 7.3.3.17.0-1) and **Filter Handling Conditions** (table 7.3.3.17.0-2). **Matching Conditions** are evaluated against resources and, when true, determine the matched resources. The **Filter Handling Conditions** provide additional input applied to the matched resource set to determine the filtering result (e.g. maximum number of resources to be included in the filtering result). The filtering result may be composed of zero or more resources and shall be used as described in clauses 7.3.3.2 and 7.3.3.14. Table 7.3.3.17.0‑1 summarizes the various filter criteria and conditions. Each row in the table represents a different filter condition type.

If no matching conditions are present, the resource is matched.

If multiple matching conditions of the same type (i.e. same condition tag) are present in the ***Filter Criteria*** parameter, these shall be combined by applying logical OR operation. This applies to the condition tags labels, resourceType, contentType or attribute with multiplicity n > 1.

If multiple matching conditions of different types (i.e. different condition tags) are present in the ***Filter Criteria*** parameter, then the combined condition shall be derived by applying the logical operation specified by the ***filterOperation***condition. By default logical AND operation shall be used if the ***filterOperation***condition is not present.

EXAMPLE:

1. ***labels***=floor1, ***stateTagSmaller***=3 will match if both conditions are true [default AND when ***filterOperation*** is not specified]
2. ***labels***=floor1, ***stateTagSmaller***=3, ***filterOperation***=1 will match if both conditions are true
3. ***labels***=floor1, ***stateTagSmaller***=3, ***filterOperation***=2 will match if either condition is true
4. ***labels***=floor1, ***labels***=floor2, ***filterOperation***=1 will match if either condition is true [***filterOperation*** has no effect when all condition tags are the same]
5. ***labels***=floor1, ***stateTagSmaller***=3, ***labels***=floor2, ***filterOperation****=2* will match if any of these conditions are true resource has [labels with value "floor1" OR "floor2"] OR stateTagSmaller than 3
6. ***labels***=floor1, ***stateTagSmaller***=3, ***filterOperation***=3 will match if one condition is true and the other condition is false

Table 7.3.3.17.0‑1: Summary of Filter Matching Conditions

| Condition Tag | Multiplicity | Targeted Resource Attribute | Matching Condition |
| --- | --- | --- | --- |
| createdBefore | 0..1 | creationTime | creationTime < createdBefore, see clause 7.3.3.17.1 |
| createdAfter | 0..1 | createdAfter ≤ creationTime, see clause 7.3.3.17.1 |
| unmodifiedSince | 0..1 | lastModifiedTime | lastModifiedTime < unmodifiedSince, see clause 7.3.3.17.2 |
| modifiedSince | 0..1 | unmodifiedSince ≤ lastModifiedTime, see clause 7.3.3.17.2 |
| stateTagSmaller | 0..1 | stateTag | stateTag < stateTagSmaller, see clause 7.3.3.17.3 |
| stateTagBigger | 0..1 | stateTagBigger ≤ stateTag, see clause 7.3.3.17.3 |
| expireBefore | 0..1 | expirationTime | expirationTime < expireBefore, see clause 7.3.3.17.4 |
| expireAfter | 0..1 | expireAfter ≤ expirationTime, see clause 7.3.3.17.4 |
| labels | 0..1 | labels | see clause 7.3.3.17.5 |
| childLabels | 0..1 | see clause 7.3.3.17.5 |
| parentLabels | 0..1 | see clause 7.3.3.17.5 |
| resourceType | 0..1 | resourceType | see clause 7.3.3.17.6 |
| childResourceType | 0..1 | see clause 7.3.3.17.6 |
| parentResourceType | 0..1 | see clause 7.3.3.17.6 |
| sizeBelow | 0..1 | contentSize | contentSize < sizeBelow, see clause 7.3.3.17.7 |
| sizeAbove | 0..1 | sizeAbove ≤ contentSize, see clause 7.3.3.17.7 |
| typeOfContent | 0..n | contentInfo | matched with typeOfContent component in contentInfo, see clause 7.3.3.17.8 |
| attribute | 0..n | (variable) | name and value of Filter Criteria attribute matches resource attribute, see clause 7.3.3.17.9 |
| childAttribute | 0..n | (variable) | name and value of Filter Criteria attribute matches resource attribute, see clause 7.3.3.17.9 |
| parentAttribute | 0..n | (variable) | name and value of Filter Criteria attribute matches resource attribute, see clause 7.3.3.17.9 |
| semanticsFilter | 0..n | (not applicable) | Filtering conditions expressed in SPARQL [i.6]. These are applicable to the descriptor attribute of <semanticDescriptor> children associated with discoverable resources. When multiple semanticsFilter elements are provided, the matching condition is fulfilled if any of the individual conditions is matched |
| contentFilterSyntax | 0..1 | (not applicable) | Indicates the Identifier for syntax to be applied for content-based discovery |
| contentFilterQuery | 0..1 | content | The query string shall be specified when contentFilterSyntax parameter is present.  See clause 7.3.3.17.13 for applicable syntax for content-based discovery |
| geoQuery | 0..1 | location |  |

**Table 7.3.3.17.0‑2: Summary of Filter Handling Conditions**

|  |  |  |
| --- | --- | --- |
| **Condition Tag** | **Multiplicity** | **Usage** |
| filterUsage | 0..1 | Indicator specifying the kind of filtering to be performed |
| limit | 0..1 | Constraint on maximum number of resources to be returned, see clause 7.3.3.17.10 |
| filterOperation | 0..1 | Indicates the logical operation (AND/OR) to be used for different conditions. The default value is logical AND. |
| level | 0..1 | Constraint on maximum number of levels in the resource tree that the operation shall span. It is permitted if the operation is Retrieve. It is only permitted for other operations if filterUsage is “Discovery-based Operation”. See clause 7.3.3.17.14. |
| offset | 0..1 | The number of direct child and descendant resources the Hosting CSE shall skip over and not process. It is permitted only on retrieve requests. See clause 7.3.3.17.15. |
| applyRelativePath | 0..1 | A resource tree relative path (e.g. /tempContainer/la) which applies after all the matching conditions have been used (i.e. a matching result has been obtained). It is permitted only if filterUsage is “Discovery” or “Discovery-based Operation”. See clause 7.3.3.17.17. |

### -----------------------End of change 9---------------------------------------------

### -----------------------Start of change 10-------------------------------------------

### 7.5.2 Elements contained in the Content primitive parameter

Clauses 7.2.1.1 and 7.2.1.2 enumerate the forms that the ***Content*** primitive parameter takes in various Request and Response cases. Note that the ***Content*** primitive parameter is denoted as primitiveContent in both CDT-requestPrimitive-v4\_1\_0.xsd and CDT-responsePrimitive-v4\_1\_0.xsd.

This clause details the Objects (elements) used in some of these cases. in the tables below.

The following elements are defined for use in the ***Content*** parameter of a request:

Table 7.5.2‑1: Elements used for request content

|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Applicable Operations** | **Data Type** | **Defined in** |
| m2m:<resourceType>  {other namespace identifier}:<resourceType> | C U | See element declaration | CDT-<resourceType>-v4\_1\_0.xsd |
| m2m:notification | N | m2m:notification | CDT-notification-v4\_1\_0.xsd |
| m2m:aggregatedNotification | N | m2m:aggregatedNotification | CDT-notification-v4\_1\_0.xsd |
| m2m:securityInfo | N | m2m:securityInfo | CDT-notification-v4\_1\_0.xsd |
| m2m:attributeList | R | m2m:attributeList | CDT-requestPrimitive-v4\_1\_0.xsd |
| m2m:responsePrimitive | N | Anonymous data type defined in the responsePrimitive declaration | CDT-responsePrimitive-v4\_1\_0.xsd |

The following elements are defined for use in the ***Content*** parameter of a response sent in reply to a request message with ***Operation*** and ***Result Content*** (rcn) parameters as given in the column "Applicable Operations" (the settings of the ***Result Content*** parameters are defined in clause 6.3.4.2.7; NP means the rcn parameter is not present).

Table 7.5.2‑2: Elements used for response content

|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Applicable Operations/rcn** | **Data Type** | **Element is Defined in** |
| m2m:<resourceType>  {other namespace identifier}:<resourceType>  See note 6 | C/1,9,NP  R/1,4,5,6,7,8,NP  U/1,9,NP  D/1,4,5,6,8  See note 1 | See element declaration | CDT-<resourceType>-v4\_1\_0.xsd |
| m2m:resource | C/3 | m2m:resourceWrapper | CDT-responsePrimitive-v4\_1\_0.xsd |
| m2m:URIList | C R U D/NP  See note 2 | list of xs:anyURI | CDT-responsePrimitive-v4\_1\_0.xsd |
| m2m:resourceRefList | R/6  See note 2 | m2m:listOfChildResourceRef | CDT-responsePrimitive-v4\_1\_0.xsd |
| m2m:aggregatedResponse | C R U D  See note 3 a)  C U D/not 11  See note 3 b) | m2m:aggregatedResponse | CDT-responsePrimitive-v4\_1\_0.xsd |
| m2m:URI | C/2  See note 4 | xs:anyURI | CDT-responsePrimitive-v4\_1\_0.xsd |
| m2m:requestPrimitive | See note 7 | Anonymous data type defined in the requestPrimitive declaration | CDT-requestPrimitive-v4\_1\_0.xsd |
| m2m:debugInfo | See note 5 | xs:string | CDT-responsePrimitive-v4\_1\_0.xsd |
| m2m:securityInfo | N/NP | m2m:securityInfo | CDT-notification-v4\_1\_0.xsd |
| m2m:queryResult | R/10  See note 8 | xs:string | CDT-responsePrimitive-v4\_1\_0.xsd |
| NOTE 1: The case rcn = 7 applies to Retrieve operation only (R/7). It retrieves the original resource in case the To parameter points to an announced resource. The case R/NP applies to Retrieve operation (Non-Discovery) only.  NOTE 2: This applies to discovery or discovery-based operations only. The format of the address (structured, unstructured) depends on the ***Desired Identifier Result Type*** parameter setting (see clause 6.3.4.2.8).  NOTE 3: This applies in two distinct cases:   1. CRUD operations on a <fanOutPoint> child resource of a <group> parent resource, independent of rcn value. The ***Content*** parameter of each response primitive included in aggregatedResponse is set as given in one of the other rows of this table. 2. Discovery-based CUD operations (*filterUsage* = 4) with rcn not present or other than “discovery result resource references” .   NOTE 4: This also applies to the response ("acknowledgement") to non-blocking requests in asynchronous and synchronous modes for any CRUD operation.  NOTE 5: This is a plain text messages which can optionally be included as debugging information in error responses. The language and content of the message is determined by the Service Provider.  NOTE 6: "{other namespace identifier}" refers to a namespace other than m2m.  NOTE 7: This applies to a polling response that contains a request for polling mechanism (see clause 7.4.22.2.2).  NOTE 8: This applies to semantic query operation only. The Originator may use the Accept option to indicate which media types are acceptable for the semantic query result, e.g. application/sparql-results+xml, or application/sparql-results+json. | | | |

The XML schema definition of the ***Content*** primitive parameter (i.e. datatype m2m:primitiveContent) allows to include XML wildcard elements. An XML representation of the ***Content*** primitive parameter shall include a root element which is associated with an XSD Global Element. The root element shall be prefixed with a namespace prefix identifier (e.g. *m2m:*) specified in the associated XSD which defines the respective Global Element. The ***Content*** primitive parameter allows to include namespaces other than m2m.

### -----------------------End of changes ---------------------------------------------