|  |
| --- |
|  |

|  |
| --- |
| CHANGE REQUEST |
| Meeting ID:\* | SDS 52 |
| Source:\* | Bob Flynn, Exacta GSS, bob.flynn@exactagss.com  |
| Date:\* | 2021-12-02 |
| Reason for Change/s:\* | Companion contribution to ARC-2017-0395R03-requestAggregationForPollingChannel\_R4 |
| CR against: Release\* | Rel-4 |
| CR against: WI\* | [ ]  Active <> [ ]  MNT maintenance / < Work Item number(optional)>Is this a mirror CR? Yes [ ]  No [x] mirror CR number: (Note to Rapporteur - use latest agreed revision)[x]  STE Small Technical Enhancements / < Work Item number (optional)>Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0004 V4.7.0 |
| Clauses \* | 7.4.21,7.4.22, 8.2.3 |
| 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 |
| Other TS/TR(s) impacted | None |
| Post Freeze checking:\* | This CR contains only essential changes and corrections? YES [x]  NO [ ] This CR may break backwards compatibility with the last approved version of the TS? YES [ ]  NO [x]  |
| Template Version: January 2019 (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

Companion contribution to ARC-2017-0395R03-requestAggregationForPollingChannel\_R4

Where we add a parameter to the <pollingChannel> resource to allow an AE to specific single or aggregated/batch messages in response to the longpolling operation.

Change 1 adds procedure description to <pollingChannelURI>

Change 2 adds new attribute to <pollingChannel>

 The default is defined as “True” to maintain same behavior as earlier releases.

Change 3 adds a shortname

R01

Change 4 adds m2m:aggregatedRequest to clause 7.5.2

R02

Change default back to false (change 2)

Change 4 (7.5.2) adding a datatype for the aggregated requests, but not removing requestPrimitive (add change 5 to define new datatype m2m: aggregatedRequestPrimitives)

Change 1: Reword the highlighted text, to something like “as follows”. (7.4.22.2.2 step Recv-6.5)

Change 5+6: Propose a new definition for the data structure returned, where we use a list of request primitives (m2m:aggregatedRequestPrimitive) instead of the m2m:aggregatedRequest which partitions the request primitive into parts (some parameters are in m2m:metainformation)

#### 6.3.5.33 m2m:aggregatedResponse

Used when aggregating responses by a group.

Table 6.3.5.33‑1: Type Definition of m2m:aggregatedResponse

|  |  |  |  |
| --- | --- | --- | --- |
| Element Path | **Element Data Type**  | Multiplicity | Note |
| resourceID | xs:anyURI | 0..1 | Reference to the <request> resource that can be used to retrieve the remaining member responses |
| m2m:responsePrimitive | See Table 6.4.2‑1 for detail | 0..n | See note |
| NOTE: The element name shall contain the namespace prefix. |

#### 6.3.5.3 m2m:aggregatedRequest

Used for the*aggregatedRequest* attribute of the <delivery> resource.

Table 6.3.5.3‑1: Type Definition of m2m:aggregatedRequest

|  |  |  |  |
| --- | --- | --- | --- |
| Element Path | **Element Data Type**  | Multiplicity | Note |
| request | (anonymous) | 1..n |  |
| request/operation | m2m:operation | 1 | See clause 6.3.4.2.5 |
| request/to | xs:anyURI | 1 |  |
| request/from | m2m:ID | 1 | See clause 6.3.3 |
| request/requestIdentifier | m2m:requestID | 1 | See clause 6.3.3 |
| request/primitiveContent | m2m:primitiveContent | 0..1 | See clause 6.3.5.5 |
| request/metaInformation | m2m:metaInformation | 0..1 | See clause 6.3.5.4 |

#### 6.3.5.4 m2m:metaInformation

Used for the *metaInformation* attribute of the <request> resource, and in the m2m:aggregatedRequest data type.

Table 6.3.5.4‑1: Type Definition of m2m:metaInformation

|  |  |  |  |
| --- | --- | --- | --- |
| Element Path | **Element Data Type**  | Multiplicity | Note |
| resourceType | m2m:resourceType | 0..1 | See clause 6.3.4.2.1 |
| originatingTimestamp | m2m:timestamp | 0..1 |  |
| requestExpirationTimestamp | m2m:absRelTimestamp | 0..1 |  |
| resultExpirationTimestamp | m2m:absRelTimestamp | 0..1 |  |
| operationExecutionTime | m2m:absRelTimestamp | 0..1 |  |
| responseType | m2m:responseTypeInfo | 0..1 | See clause 6.3.4.2.6 |
| resultPersistence | m2m:absRelTimestamp | 0..1 |  |
| resultContent | m2m:resultContent | 0..1 | See clause 6.3.4.2.7 |
| eventCategory | m2m:eventCat | 0..1 | See clause 6.3.3 |
| deiveryAggregation | xs:boolean | 0..1 |  |
| groupRequestIdentifier | xs:string | 0..1 |  |
| filterCriteria | m2m:filterCriteria | 0..1 | See clause 6.3.5.8 |
| desiredIdentifierResultType | m2m:desIdResType | 0..1 | See clause 6.3.4.2.8 |
| roleIDs | List of m2m:roleID | 0..1 |  |
| tokenRequestIndicator | xs:boolean | 0..1 |  |
| tokens | List of m2m:dynAuthJWT | 0..1 |  |
| tokenIDs | List of m2m:tokenID | 0..1 |  |
| localTokenIDs | List of xs:NCName | 0..1 |  |
| groupRequestTargetMembers | List of xs:anyURI | 0..1 |  |
| groupSomecastTargetNumber | xs:positiveInteger | 0..1 |  |
| authorSignIndicator | xs:boolean | 0..1 |  |
| authorSigns | m2m:signatureList | 0..1 |  |
| authorRelIndicator | xs:boolean | 0..1 |  |
| semanticQueryIndicator | xs:boolean | 0..1 |  |
| ontologyMappingResources | m2m:listOfM2MID | 0..1 |  |
| ReleaseVersionIndicator | m2m:releaseVersion | 1 |  |
| vendorInformation | xs:string | 0..1 |  |
| primitiveProfileIdentifier | xs:anyURI | 0..1 | See Clause [7.4.72](#_7.4.72_Resource_Type) |
| M2M Service User | m2m:ID | 0..1 |  |

-------------------------------------------------- Start of Change 1---------------------------------------------------

### 7.4.22 Resource Type <pollingChannelURI>

#### 7.4.22.1 Introduction

The <pollingChannelURI> resource is the virtual child resource which is automatically generated during the parent <pollingChannel> resource creation. The detailed description can be found in clause 9.6.22 in oneM2M TS-0001 [6].

There is no data type definition for <pollingChannelURI> resource because it is a virtual resource type.

#### 7.4.22.2 <pollingChannelURI> resource specific procedures for CRUD operations

##### 7.4.22.2.0 Introduction

This clause describes <pollingChannelURI> resource specific behaviour for the Retrieve operation as a service layer long polling request. CUDN requests to the <pollingChannelURI> resource shall be rejected.

##### 7.4.22.2.1 Create

***Originator:***

The <pollingChannelURI> resource shall not be created via API.

***Receiver:***

Primitive specific operation on Recv-1.0 "Check the validity of received request primitive":

1. If the request is received, the Receiver CSE shall execute the following steps in order:
2. "Create an unsuccessful Response primitive" with the ***Response Status Code*** indicating an "OPERATION\_NOT\_ALLOWED" error.
3. "Send the Response primitive".

##### 7.4.22.2.2 Retrieve

***Originator:***

Shall execute Originator actions in clause 7.2.2.1 as a service layer long polling request. It is the Originator's responsibility to initiate this procedure after it gets long polling response either successful or unsuccessful. The Originator shall send this Retrieve request as blocking request (clause 8.2.1 in oneM2M TS-0001 [6]).

***Receiver:***

Shall execute the following steps in order and these are modifications to the generic procedure from Recv-6.3 to Recv‑6.5 in clause 7.2.2.2:

Recv-6.3 Check if the request Originator is the creator of the parent <pollingChannel> resource. If it is not the creator, the Hosting CSE shall send a response primitive with a ***Response Status Code*** indicating "ORIGINATOR\_HAS\_NO\_PRIVILEGE" error.

Recv-6.4 No change from the generic procedure.

Recv-6.5 If there is a pending request(s) to be sent to the Originator:

* Create a Response primitive, setting its Content parameter as follows:
	+ If *requestAggregation* attribute is **True** then return all pending requests in the response in a m2m: aggregatedRequestPrimitive, where the requests are listed in order of arrival. If *requestAggregation* attribute is **False** then return one of the pending request(s) in the response in a m2m:requestPrimitive.
	+ If the request includes a ***Primitive Profile Identifier*** parameter and the referenced <primitiveProfile> resource has an *applicability* attribute with the value “NOTIFICATIONS \_FROM\_CSE” the HOSTING CSE shall not apply this primitive profile to any notification requests included in the ***Content*** parameter of the Response primitive.

Else:

* Wait for a request for the Originator until the ***Request Expiration Timestamp*** of the Originator's request. If a request is available before the ***Request Expiration Timestamp*** timeout, create a Response primitive setting the ***Content*** parameter to the received pending request. Otherwise, create a response primitive with a ***Response Status Code*** indicating "REQUEST\_TIMEOUT" error.

##### 7.4.22.2.3 Update

***Originator:***

The <pollingChannelURI> resource shall not be updated via API.

***Receiver:***

Primitive specific operation on Recv-1.0 "Check the validity of received request primitive":

1. If the request is received, the Receiver CSE shall execute the following steps in order:
2. "Create an unsuccessful Response primitive" with the ***Response Status Code*** indicating an "OPERATION\_NOT\_ALLOWED" error.
3. "Send the Response primitive".

##### 7.4.22.2.4 Delete

***Originator:***

The <pollingChannelURI> resource shall not be deleted via API.

***Receiver:***

Primitive specific operation on Recv-1.0 "Check the validity of received request primitive":

1. If the request is received, the Receiver CSE shall execute the following steps in order:
2. "Create an unsuccessful Response primitive" with the ***Response Status Code*** indicating an "OPERATION\_NOT\_ALLOWED" error.
3. "Send the Response primitive".

##### 7.4.22.2.5 Notify

Request/response delivery mechanism via <pollingChannelURI> resource is depicted in the TS-0001 [6], Figure 10.2.5.12-1 (Request/response delivery via polling channel). In this procedure, the Originator is the Target AE/CSE and the Receiver is the <pollingChannelURI> Hosting CSE, respectively.

***Originator:***

This procedure follows the procedure specified in clause 7.2.2.1 with the following <pollingChannelURI> resource-specific updates.

Additional primitive specific operation on Orig-1.0:

* The Originator shall generate and populate a Notify request as described in clause 7.5.1.2.7.

***Receiver:***

The following are additional Hosting CSE procedures to the generic resource handling procedures from Recv-6.3 to Recv-6.5 in clause 7.2.2.2:

Recv-6.3 Check if the request Originator is the creator of the parent <pollingChannel> resource. If it is not the creator, the Hosting CSE shall send response primitive with a ***Response Status Code*** indicating "ORIGINATOR\_HAS\_NO\_PRIVILEGE" error.

Recv-6.4 No change from the generic procedure.

Recv-6.5 Forward the response (step 006 in Figure 10.2.5.12-1 of TS-0001 [6]), which was contained in the ***Content*** parameter of the Notify request, to the entity that sent the associated request to the Hosting CSE (Originator in the figure 10.2.5.12-1). The associated request is the request that the Hosting CSE received and forwarded to the Registree AE or CSE over the polling channel (step 002 and step 004 in the figure). The association shall be done by matching the ***Request Identifier*** parameter of the request delivered in <pollingChannelURI> Retrieve response (step 004 in the figure) and the ***Request Identifier*** parameter of the response delivered in the ***Content*** parameter in a <pollingChannelURI> Notify request (step 005 in the figure).

-------------------------------------------------- End of Change 1---------------------------------------------------

-------------------------------------------------- Start of Change 2---------------------------------------------------

### 7.4.21 Resource Type <pollingChannel>

#### 7.4.21.1 Introduction

The <pollingChannel> resource is used to perform service layer long polling when an AE/CSE cannot receive a request from other entities, however it can get a request as a response to a long polling request. Actual long polling can be performed on the <pollingChannelURI> resource which is the child resource of the <pollingChannel> resource.

The detailed description can be found in clause 9.6.21 in oneM2M TS-0001 [6].

Table 7.4.21.1‑1: Data type definition of <pollingChannel> resource

|  |  |  |
| --- | --- | --- |
| Data Type ID | File Name | Note |
| pollingChannel | CDT-pollingChannel.xsd |  |

Table 7.4.21.1‑2: Universal/Common Attributes of <pollingChannel> resource

|  |  |
| --- | --- |
| Attribute Name | Request Optionality  |
| Create | Update |
| *@resourceName* | O | NP |
| *resourceType* | NP | NP |
| *resourceID* | NP | NP |
| *parentID* | NP | NP |
| *creationTime* | NP | NP |
| *lastModifiedTime* | NP | NP |
| *labels* | O | O |
| *expirationTime* | O | O |

Table 7.4.21.1‑3: Child resources of <pollingChannel> resource

|  |  |  |  |
| --- | --- | --- | --- |
| Child Resource Type | Name | Multiplicity | Ref. to Resource Type Definition |
| <pollingChannelURI> | pcu | 1 | Clause 7.4.22 |
| <transaction> | [variable] | 0..n | Clause 7.4.61 |

Table 7.4.57.1‑3: Resource Specific Attributes of <pollingChannel> resource

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Request Optionality  | Data Type | Default Value and Constraints |
| Create | Update |
| *requestAggregation* | O | O | xs:boolean | False |

#### 7.4.21.2 <pollingChannel> resource specific procedures for CRUD operations

##### 7.4.21.2.0 Introduction

This clause describes <pollingChannel> resource specific behaviour for CRUD operations.

##### 7.4.21.2.1 Create

***Originator:***

No change from the generic procedures in clause 7.2.2.1.

***Receiver:***

Same as the generic procedures in clause 7.2.2.2 except one addition:

* After Recv-6.3 procedure, the Hosting CSE shall check if the Originator ID is the same as the AE-ID or CSE-ID of the target <AE> resource or <remoteCSE> resource, respectively. If the check fails, then the Hosting CSE shall return a response primitive with a ***Response Status Code*** indicating "ORIGINATOR\_HAS\_NO\_PRIVILEGE" error.

##### 7.4.21.2.2 Retrieve

***Originator:***

No change from the generic procedures in clause 7.2.2.1.

***Receiver:***

Same as the generic procedures in clause 7.2.2.2 except one addition:

* After Recv-6.3 procedure, the Hosting CSE shall check if the Originator ID is the same as the AE-ID or CSE-ID of the target <AE> resource or <remoteCSE> resource, respectively. If the check fails, then the Hosting CSE shall return a response primitive with a ***Response Status Code*** indicating "ORIGINATOR\_HAS\_NO\_PRIVILEGE" error.

##### 7.4.21.2.3 Update

***Originator:***

No change from the generic procedures in clause 7.2.2.1.

***Receiver:***

Same as the generic procedures in clause 7.2.2.2 except one addition:

* After Recv-6.3 procedure, the Hosting CSE shall check if the Originator ID is the same as the AE-ID or CSE-ID of the target <AE> resource or <remoteCSE> resource, respectively. If the check fails, then the Hosting CSE shall return a response primitive with a ***Response Status Code*** indicating "ORIGINATOR\_HAS\_NO\_PRIVILEGE" error.

##### 7.4.21.2.4 Delete

***Originator:***

No change from the generic procedures in clause 7.2.2.1.

***Receiver:***

Same as the generic procedures in clause 7.2.2.2 except one addition:

* After Recv-6.3 procedure, the Hosting CSE shall check if the Originator ID is the same as the AE-ID or CSE-ID of the target <AE> resource or <remoteCSE> resource, respectively. If the check fails, then the Hosting CSE shall return a response primitive with a ***Response Status Code*** indicating "ORIGINATOR\_HAS\_NO\_PRIVILEGE" error.

-------------------------------------------------- End of Change 2---------------------------------------------------

-------------------------------------------------- Start of Change 3---------------------------------------------------

### 8.2.3 Resource attributes

Table 8.2.3‑1: Resource attribute short names (1/6)

| Attribute Name | Occurs in | Short Name |
| --- | --- | --- |
| *requestAggregation* | pollingChannel | *rqag* |

-------------------------------------------------- End of Change 3---------------------------------------------------

-------------------------------------------------- Start of Change 4---------------------------------------------------

### 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.xsd and CDT-responsePrimitive.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>.xsd |
| m2m:notification | N | m2m:notification | CDT-notification.xsd |
| m2m:aggregatedNotification | N | m2m:aggregatedNotification | CDT-notification.xsd |
| m2m:securityInfo | N | m2m:securityInfo | CDT-notification.xsd |
| m2m:attributeList | R | m2m:attributeList | CDT-requestPrimitive.xsd |
| m2m:responsePrimitive | N | Anonymous data type defined in the responsePrimitive declaration | CDT-responsePrimitive.xsd |
| m2m:timeSyncBeaconNotification | N | m2m:timeSyncBeaconNotification | CDT-timeSyncBeacon.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,NPR/1,4,5,6,7,8,NPU/1,9,NPD/1,4,5,6,8See note 1 | See element declaration | CDT-<resourceType>.xsd |
| m2m:resource | C/3 | m2m:resourceWrapper | CDT-responsePrimitive.xsd |
| m2m:URIList | C R U D/11,NPSee note 2 | list of xs:anyURI | CDT-responsePrimitive.xsd |
| m2m:resourceRefList | C R U D/6See note 2 | m2m:listOfChildResourceRef  | CDT-responsePrimitive.xsd |
| m2m:aggregatedResponse | C R U DSee note 3 a)C U D/not 11See note 3 b) | m2m:aggregatedResponse | CDT-responsePrimitive.xsd |
| m2m:URI | C/2See note 4 | xs:anyURI | CDT-responsePrimitive.xsd |
| m2m:requestPrimitive | See note 7 | Anonymous data type defined in the requestPrimitive declaration | CDT-requestPrimitive.xsd |
| m2m: aggregatedRequestPrimitives | See note 7 | m2m: aggregatedRequestPrimitives | CDT-requestPrimitive.xsd |
| m2m:debugInfo | See note 5 | xs:string | CDT-responsePrimitive.xsd |
| m2m:securityInfo | N/NP | m2m:securityInfo | CDT-notification.xsd |
| m2m:queryResult | R/10See note 8 | xs:string | CDT-responsePrimitive.xsd |
| m2m:permissions | R/12See note 9 | m2m:resourcePermissions | CDT-responsePrimitive.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 rcn values listed for Retrieve (R) apply to retrieve operations when filterUsage is conditional retrieval (2).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 not equal to 11 (“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 one or more requests 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.NOTE 9: The case rcn=12 applies to a Retrieve operation only and contains the access control privileges that the Originator has on the targeted resource and/or any child/descendant(s) (see clause 7.2.1.2). |

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 Change 4---------------------------------------------------

-------------------------------------------------- Start of Change 5---------------------------------------------------

#### 6.3.5.xx m2m:aggregatedRequestPrimitives

Used when a batch of multiple requests may be sent to an AE or CSE.

Table 6.3.5.xx‑1: Type Definition of m2m:aggregatedRequestPrimitives

|  |  |  |  |
| --- | --- | --- | --- |
| Element Path | **Element Data Type**  | Multiplicity | Note |
|  |  |  |  |
| m2m:requestPrimitive | See Table 6.4.1‑1 for detail | 1..n | See note |
| NOTE: The element name shall contain the namespace prefix. |

-------------------------------------------------- End of Change 5---------------------------------------------------

-------------------------------------------------- Start of Change 6---------------------------------------------------

### 8.2.5 Complex data types members

In protocol bindings complex data type member names shall be translated into short names of Table 8.2.5-1.

Table 8.2.5‑1: Complex data type member short names

|  |  |  |
| --- | --- | --- |
| aggregatedResponse | Response Primitive Content | ***agr*** |
| aggregatedRequestPrimitives | Response Primitive Content | ***agrp*** |
| resource | Response Primitive Content | ***rce*** |
| URIList | Response Primitive Content | ***uril*** |
| debugInfo | Response Primitive Content | ***dbg*** |
| queryResult | Response Primitive Content | ***qres*** |

-------------------------------------------------- End of Change 6---------------------------------------------------