|  |
| --- |
|  |

|  |
| --- |
| CHANGE REQUEST |
| Meeting ID:\* | SDS 50 |
| Source:\* | Siddharth Trikha, strikha@cdot.in Poornima Shandilya poornima@cdot.in |
| Date:\* | 2021-05-20 |
| Reason for Change/s:\* | SPARQL query error status code |
| CR against: Release\* | Rel-3 |
| CR against: WI\* | [ ]  Active < WI-0077> [x]  MNT maintenance / < Work Item number(optional)>Is this a mirror CR? Yes [x]  No [ ] mirror CR number: [ ]  STE Small Technical Enhancements / < Work Item number (optional)>Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0004 v3.21.0 |
| Clauses \* | 6.6.3.5, 7.3.3.18, 7.3.3.19, 7.4.34.2, 7.4.35.2, 7.4.47.2, 7.4.49.2 |
| Type of change: \* | [ ]  Editorial change[x]  Bug Fix or Correction[ ]  Change to existing feature or functionality[ ]  New feature or functionalityOnly ONE of the above shall be ticked |
| 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

Whenever the originator sends an non-valid SPARQL query in a oneM2M Request, the status code returned by the CSE should be changed to a semantic specific error code instead of the generic BAD\_REQUEST.

This contribution adds a Status code for Invalid SPARQL query and replaces BAD\_REQUEST with the new status code.

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

#### 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 |
| 4128 | SERVICE\_SUBSCRIPTION\_NOT\_ESTABLISHED |
| 41xx | INVALID\_SPARQL\_QUERY |

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

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

#### 7.3.3.18 Semantic resource discovery

##### 7.3.3.18.0 Introduction

Semantic resource discovery is used to find resources in a CSE based on the semantic descriptions contained in the *descriptor* attribute of <*semanticDescriptor*> resources. Since an overall semantic description (forming a graph [Error: Reference source not found]) may be distributed across a set of <semanticDescriptor> resources, the semantic descriptions have to be retrieved (before or as needed) during the execution of the discovery request.

Semantic resource discovery is initiated by sending a Retrieve request with the discovery criteria in the ***semanticsFilter***filter condition(s) with two alternatives:

1. Targeting a <*semanticFanOutPoint>* virtual resource, see clause Error: Reference source not found.
2. Targeting a resource other than <semanticFanOutPoint>. In this alternative the semantic resource discovery request procedure shall be comprised of the following actions:

***Originator:***

The Originator shall follow the steps from Orig-1.0 to Orig-6.0 specified in clause Error: Reference source not found Generic Resource Request Procedure for Originator.

In addition to Orig-1.0, the following steps shall be performed.

The *To* parameter in the Retrieve Request shall indicate the root of where the semantic discovery begins.

The *filterCriteria* of the Retrieve Request shall include the *filterUsage* parameter configured as "discovery" and the ***semanticsFilter*** filter condition.

***Receiver:***

The Receiver shall follow the steps from Recv-1.0 to Recv-7.0 specified in clause Error: Reference source not found Generic Resource Request Procedure for Receiver.

After Recv-1.0 "Check the validity of received request primitive": check that the syntax of the ***semanticsFilter*** corresponds to a valid SPARQL query request [Error: Reference source not found]. If the ***semanticsFilter*** content does not correspond to a valid SPARQL query request, the Receiver shall generate a ***Response Status Code*** indicating a "INVALID\_SPARQL\_QUERY" error.

The Hosting CSE shall follow the steps from Recv-1.0 to Recv-6.2 specified in clause Error: Reference source not found.The Hosting CSE shall not perform steps from Recv-6.3 to Recv-6.6 and perform the following steps instead:

1. The Hosting CSE shall find the <semanticDescriptor> resource(s) to which the Originator has "Discover" access right, under the addressed resource.

a) If the *relatedSemantics* attribute does not exist, the "Annotation-based method" (using *resourceDescriptorLink)* detailed in clause shall be used.

b) If the *relatedSemantics* attribute exists the "Resource link-based method" (using the *relatedSemantics attribute*)detailed in clause shall be used.

The Hosting CSE shall perform Recv-6.7 "Create a success response" where the Response shall include the resources matched based on the SPARQL engine result.

##### 7.3.3.18.1 Annotation-based method

In the annotation-based method, related <semanticDescriptor> resources are identified within the RDF semantic description itself using a special annotation property called *m2m:resourceDescriptorLink*. This property points to another <semanticDescriptor> resource which may contain relevant information for matching the semantic filter. Whenever, during the execution of the SPARQL request (on the semantic description in the *descriptor* attribute of the <semanticDescriptor>) such an annotation property is found, the execution is halted, the content of the *descriptor* attribute of the referred to <semanticDescriptor> is retrieved, and the execution is continued on the combined content of the already present and the just retrieved semantic information.

##### 7.3.3.18.2 Resource link-based method

In this option, the *relatedSemantics* attribute contains the list of <semanticDescriptor> resources which shall be retrieved for the purpose of creating the overall graph against which the SPARQL request is executed.

The Hosting CSE retrieves the <semanticDescriptor> child resource of the request target and the addresses provided in the *relatedSemantics* attribute. For each address from the *relatedSemantics* list the Hosting CSE:

checks that the Originator has "Discover" access rights, and the existence of the addressed resource;

retrieves the description in the *descriptor* attribute under the addressed resource.

The Hosting CSE shall aggregate all the retrieved descriptors and deliver the content for SPARQL request processing, along with the ***semanticsFilter*** content.

NOTE: In the resource link-based method, no actions need to be performed during the execution of the SPARQL request if the notation property *onem2m:resourceDescriptorLink* is encountered.

Afterwards, the Hosting CSE performs Recv-6.7 "Create a success response" where the Response shall include the resources matched based on the SPARQL engine result.

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

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

#### 7.3.3.19 Semantic query

##### 7.3.3.19.0 Introduction

Semantic queries enable the retrieval of both explicitly and implicitly derived information based on syntactic, semantic and structural information contained in data (such as RDF data). The result of a semantic query is the semantic information/knowledge for answering/matching the query. Note that, in the following descriptions, the general term semantic resource is used to refer to <semanticDescriptor> resources and any other future resources containing semantic information.

For a given semantic query, it needs to be executed on a set of RDF triples (called the "RDF data basis"), which may be distributed in the resource tree and stored in different semantic resources. The Receiver shall perform semantic graph scoping, which is the process of establishing the "query scope" for this semantic query in order to build its RDF data basis. The following two approaches may be used to decide the semantic query scope of a semantic query:

Approach-1: The scope of the semantic query is provided implicitly.

Approach-2: The scope of the semantic query is provided explicitly

##### 7.3.3.19.1 Approach-1: Semantic query with implicit scope

In Approach-1, a semantic query request message targets any resource (i.e. as specified by the ***To*** parameter) and the semantic query shall be executed relative to this target resource, similarly to other request messages. The scope of the semantic query is formed through the aggregation of the semantic contents of the target resource's descendants. All the contents of semantic resource descendants of the target resource shall form the RDF data basis for this semantic query to be executed on. In this alternative, the semantic query procedure shall be comprised of the following actions:

***Originator:***

The Originator shall follow the steps from Orig-1.0 to Orig-6.0 specified in clause Error: Reference source not found Generic Resource Request Procedure for Originator.

In addition to Orig-1.0, the following steps shall be performed.

The ***To*** parameter in the Retrieve Request shall define the scope of this semantic query as mentioned earlier.

The Retrieve Request shall include the following parameters:

1. the ***Semantic Query Indicator***, which is set to true;

***filterCriteria*** of the Retrieve Request shall include the ***semanticsFilter*** condition tag; and

the parameter Result Content shall be set to "semantic content" to indicate that the response message shall contain the result of a semantic query.

***Receiver:***

The Receiver shall follow the steps from Recv-1.0 to Recv-7.0 specified in clause Error: Reference source not found Generic Resource Request Procedure for Receiver.

After Recv-1.0 "Check the validity of received request primitive": check that the syntax of the ***semanticsFilter*** corresponds to a valid SPARQL query request [Error: Reference source not found]. If the ***semanticsFilter*** content does not correspond to a valid SPARQL query request, the Receiver shall generate a ***Response Status Code*** indicating a "INVALID\_SPARQL\_QUERY" error.

The Hosting CSE shall follow the steps from Recv-1.0 to Recv-6.2 specified in clause Error: Reference source not found. The Hosting CSE shall not perform steps from Recv-6.3 to Recv-6.6 and perform the following steps instead:

1. The Hosting CSE shall find the semantic resources to which the Originator has "RETRIEVE" access right, under the addressed resource as specified by the ***To*** parameter.
2. Aggregate the semantic resources and deliver the content for SPARQL processing, along with the ***semanticsFilter*** content.
3. Wait for a SPARQL processing response.
4. Perform Recv-6.7 "Create a success response" where the Response shall include the SPARQL processing result, which is the semantic query result to be returned.
5. Perform Recv-6.8 and the procedure is terminated.

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

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

#### 7.4.34.2 <semanticDescriptor> resource specific procedures for CRUD operations

##### 7.4.34.2.0 Introduction

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

##### 7.4.34.2.1 Create

***Originator:***

No change from the generic procedures in clause Error: Reference source not found.

***Receiver:***

No change from the generic procedures in clause Error: Reference source not found with the following exception:

1. Primitive specific operation on Recv-6.4 "Check validity of resource representation for the given resource type":

a) The Hosting CSE shall check that the *descriptor* attribute conforms to the syntax defined by the *descriptorRepresentation* attribute.

b) If the *descriptor* attribute does not conform, the Hosting CSE shall reject the request with a ***Response Status Code*** indicating a "BAD\_REQUEST" error .

c) The Hosting CSE shall reject the request with a ***Response Status Code*** indicating a "BAD\_REQUEST" error if the *descriptorRepresentation* attribute is set to "IRI".

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

a) The Hosting CSE shall set the *validationEnable* attribute of the <semanticDescriptor> resource based on the value provided in the request and its local policy. Note that the local policy may override the suggested value provided in the request from the originator to enforce or disable the following semantic validation procedures. There are different cases depending on how the local policy is configured (which is out of the scope of the present document) and whether/how the *validationEnable* attribute is provided in the request:

*validationEnable* attribute is not present if it was not provided in the request or if the local policy does not allow for the *validationEnable* attribute;

*validationEnable* attribute is set to true or false according to the local policy no matter how the value is provided in the request;

*validationEnable* attribute is set to true or false according to the value provided in the request.

b) If the *validationEnable* attribute is set as true, the hosting CSE shall perform the semantic validation process in the following steps according to clause 7.10.2 in oneM2M TS-0034 [Error: Reference source not found]. Otherwise, skip the following steps.

c) Check if the addressed *<semanticDescriptor>* resource is linked to other *<semanticDescriptor>* resources on a remote CSE by the *relatedSemantics* attribute or by triples with annotation property *m2m:resourceDescriptorLink* in *descriptor* attribute. This process shall consider the recursive links.

If yes, the Hosting CSE shall generate an Update request primitive with itself as the Originator and with the ***Content*** parameter set to the addressed *<semanticDescriptor>* resource representation, and send it to the <semanticValidation> virtual resource URI on the CSE which hosts the referenced ontology (following the *ontologyRef* attribute) of the addressed *<semanticDescriptor>* resource (see details in clause 7.4.48.2.3). After receiving the response primitive, i.e. the validation result, go to step k. If no response primitive was received due to time-out or other exceptional cases, the hosting CSE shall generate a ***Response Status Code*** indicating a "TARGET\_NOT\_REACHABLE" error.

If no, perform the following steps.

d) Access the semantic triples from the *descriptor* attribute of the received <semanticDescriptor> resource.

e) Access the ontology referenced in the *ontologyRef* attribute of the received <semanticDescriptor> resource.

If the ontology referenced by the *ontologyRef* attribute is an external ontology, not locally hosted by the Hosting CSE, the Hosting CSE shall retrieve it using the corresponding protocol and identifier information specified in the *ontologyRef* attribute.

If the referenced ontology cannot be retrieved within a reasonable time (as defined by a local policy), the Hosting CSE shall generate a ***Response Status Code*** indicating an "ONTOLOGY\_NOT\_AVAILABLE" error.

f) Retrieve any local linked <semanticDescriptor> resources of the received <semanticDescriptor> resource following the URI(s) in the *relatedSemantics* attribute (if it exists) and the URI(s) in the triples with annotation property m2m:resourceDescriptorLink (if there are any).

Repeat this step recursively to Retrieve any further local linked <semanticDescriptor> resources.

If the local linked <semanticDescriptor> resources cannot be retrieved within a reasonable time (which is subject to a local policy), the Hosting CSE shall generate a ***Response Status Code*** indicating a "LINKED\_SEMANTICS\_NOT\_AVAILABLE" error.

g) Retrieve the semantic triples from the *descriptor* attribute of the local linked <semanticDescriptor> resource.

h) Retrieve the referenced ontologies of the local linked <semanticDescriptor> resources following the URI(s) in *ontologyRef* attribute of the linked <semanticDescriptor> resources; If the referenced ontologies cannot be retrieved within a reasonable time (as defined by a local policy), the Hosting CSE shall generate a ***Response Status Code*** indicating an "ONTOLOGY\_NOT\_AVAILABLE" error.

i) Combine all the semantic triples of the addressed and local linked <semanticDescriptor> resources as the set of semantic triples to be validated, and combine all the referenced ontologies as the set of ontologies to validate the semantic triples against.

j) Check all the aspects of semantic validation according to clause 7.10.3 in oneM2M TS-0034 [Error: Reference source not found] based upon the semantic triples and referenced ontology. If any problem occurs, the Hosting CSE shall generate a ***Response Status Code*** indicating an "INVALID\_SEMANTICS" error.

k) After the semantic validation process, the Hosting CSE shall set the *semanticValidated* attribute of the addressed <semanticDescriptor> resource according to the validation result (i.e. set to true if the no error occurs until now, otherwise false).

l) Based on its local policy, the Hosting CSE may also update the value of the *semanticValidated* attributes of the local linked <semanticDescriptor> resources according to the validation result.

##### 7.4.34.2.2 Retrieve

***Originator:***

No change from the generic procedures in clause Error: Reference source not found.

***Receiver:***

No change from the generic procedures in clause Error: Reference source not found with the following exception:

The *semanticOpExec* attribute is never returned in the response.

##### 7.4.34.2.3 Update

***Originator:***

No change from the generic procedures in clause Error: Reference source not found with the following exception:

The *descriptor* attribute can be updated using SPARQL as follows:

 Primitive specific operation on Orig-1.0 "Compose Request primitive": The originator creates a request to update the *semanticOpExec* attribute. The value of this attribute is set to a SPARQL request that includes INSERT, DELETE, or DELETE/INSERT with conditional SPARQL statements as defined in the SPARQL query language [Error: Reference source not found].

***Receiver:***

No change from the generic procedures in clause Error: Reference source not found with the following exceptions:

1. Primitive specific operation on Recv-6.4 "Check validity of resource representation for the given resource type":

a) If both *semanticOpExec* and *descriptor* attributes exist, the Receiver shall generate a ***Response Status Code*** indicating a "BAD\_REQUEST" error.

b) If *semanticOpExec* attribute exists in the Request check that the syntax of its content corresponds to a valid SPARQL query request [Error: Reference source not found]. If the content does not correspond to a valid SPARQL query request, the Receiver shall generate a ***Response Status Code*** indicating a "INVALID\_SPARQL\_QUERY" error.

c) If the *descriptor* attribute exists in the Request, check that the syntax of its content conforms to the syntax defined by the *descriptorRepresentation* attribute. If the content does not conform, the Receiver shall reject the request with a ***Response Status Code*** indicating a "BAD\_REQUEST" error.

d) The Hosting CSE shall reject the request with a ***Response Status Code*** indicating a "BAD\_REQUEST" error if the *descriptorRepresentation* attribute is set to "IRI".

Primitive specific operation on Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed" in addition:

a) If *semanticOpExec* attribute exists in the Request, the Hosting CSE shall update the semantic triples in the *descriptor* attribute according to SPARQL update request in the *semanticOpExec* attribute. If the SPARQL update request cannot be executed, the Hosting CSE shall "create an unsuccessful Response primitive" with the ***Response Status Code*** indicating "SPARQL\_UPDATE\_ERROR", otherwise proceed to step Recv-6.6.

b) The hosting CSE shall set the *validationEnable* attribute of the addressed <semanticDescriptor> resource based on the value provided in the request and its local policy. Note that the local policy may override the suggested value provided in the request from the originator to enforce or disable the following semantic validation procedures. There are different cases depending on how the local policy is configured (which is out of the scope of the present document) and whether/how the *validationEnable* attribute is provided in the request:

no change to the existing *validationEnable* attribute if it is not provided in the request;

*validationEnable* attribute is not present if the local policy does not allow for the *validationEnable* attribute;

*validationEnable* attribute is set to true or false according to the local policy no matter how the value is provided in the request;

*validationEnable* attribute is set to true or false according to the value provided in the request.

c) The hosting CSE shall perform steps 2b-2l as specified in clause 7.4.34.2.1.

d) If *validationEnable* attribute is changed from true to false, then the hosting CSE shall set the *semanticValidated* attribute of the addressed <semanticDescriptor> resource as false.

-------------------------------------------------- End of Change 4---------------------------------------

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

#### 7.4.35.2 <semanticFanOutPoint> resource specific procedures for CRUD operations

##### 7.4.35.2.0 Introduction

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

##### 7.4.35.2.1 Create

***Originator:***

The <semanticFanOutPoint> resource shall not support Create operations via API.

***Receiver:***

Primitive specific operation on Recv-1.0 "Check the syntax of received message":

If the request is received, the Receiver CSE shall execute the following steps in order:

1. "Create an unsuccessful Response primitive" with the ***Response Status Code*** indicating "OPERATION\_NOT\_ALLOWED" error.
2. "Send the Response primitive".

##### 7.4.35.2.2 Retrieve

***Originator:***

No primitive specific operations.

***Receiver:***

The Receiver shall follow the steps from Recv-1.0 to Recv-6.2 specified in clause Error: Reference source not found Generic Resource Request Procedure for Receiver, with the following primitive specific operations:

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

1. Check that the syntax of the ***semanticsFilter*** corresponds to a valid SPARQL query request [Error: Reference source not found]. If the ***semanticsFilter*** does not correspond to a valid SPARQL query request, the Receiver shall generate a ***Response Status Code*** indicating a "INVALID\_SPARQL\_QUERY" error.
2. If the ***Semantic Query Indicator*** parameter included in the request message is set to true, the request shall be processed as a semantic query. Otherwise, the request shall be processed as a semantic resource discovery.

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

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

#### 7.4.47.2 <ontology> resource specific procedures for CRUD operations

##### 7.4.47.2.0 Introduction

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

##### 7.4.47.2.1 Create

***Originator:***

No change from the generic procedures in clause Error: Reference source not found.

***Receiver:***

No change from the generic procedures in clause Error: Reference source not found with the following exception:

1. Primitive specific operation on Recv-6.4 "Check validity of resource representation for the given resource type":

a) The Hosting CSE shall check that the *ontologyContent* attribute conforms to the syntax defined by the *ontologyFormat* attribute.

b) If the *ontologyContent* attribute does not conform, the Hosting CSE shall reject the request with a ***Response Status Code*** indicating a "BAD\_REQUEST" error.

##### 7.4.47.2.2 Retrieve

***Originator:***

No change from the generic procedures in clause Error: Reference source not found.

***Receiver:***

No change from the generic procedures in clause Error: Reference source not found.

##### 7.4.47.2.3 Update

***Originator:***

No change from the generic procedures in clause Error: Reference source not found with the following exception:

Primitive specific operation on Orig-1.0 "Compose Request primitive": The originator creates a request to update the *semanticOpExec* attribute. The value of this attribute is set to a SPARQL request that includes INSERT, DELETE, or DELETE/INSERT with conditional SPARQL statements as defined in the SPARQL query language [Error: Reference source not found].

***Receiver:***

No change from the generic procedures in clause Error: Reference source not found with the following exceptions:

1. Primitive specific operation on Recv-6.4 "Check validity of resource representation for the given resource type":

a) If both *semanticOpExec* and *ontologyContent* attributes exist, the Receiver shall generate a ***Response Status Code*** indicating a "BAD\_REQUEST" error.

b) If the *semanticOpExec* attribute exists in the Request check that the syntax of its content corresponds to a valid SPARQL query request [Error: Reference source not found]. If the content does not correspond to a valid SPARQL query request, the Receiver shall reject the Request with a ***Response Status Code*** indicating a "INVALID\_SPARQL\_QUERY" error.

c) If the *ontologyContent* attribute exists in the Request, check that the syntax of its content conforms to the syntax specified by the *ontologyFormat* attribute. If the content does not conform, the Receiver shall reject the Request with a ***Response Status Code*** indicating a "BAD\_REQUEST" error.

Primitive specific operation on Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed" in addition:

a) If the *semanticOpExec* attribute exists in the Request, the Hosting CSE shall update the semantic triples in the *ontologyContent* attribute according to the SPARQL update request in the s*emanticOpExec* attribute. If the SPARQL update request cannot be executed, the Hosting CSE shall "create an unsuccessful Response primitive" with the ***Response Status Code*** indicating "SPARQL\_UPDATE\_ERROR", otherwise proceed to step Recv-6.6.

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

## ---------------------------------------- Start of Change 7-------------------------------------

#### 7.4.49.2 <semanticMashupJobProfile> resource specific procedures for CRUD operations

##### 7.4.49.2.0 Introduction

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

##### 7.4.49.2.1 Create

***Originator:***

No change from the generic procedures in clause Error: Reference source not found.

***Receiver:***

No change from the generic procedures in clause 7.2.2.2 with the following exception:

1. Primitive specific operation on Recv-6.4 "Check validity of resource representation for the given resource type":

a) The Hosting CSE shall check that the *inputDescriptor, outputDescriptor* and *functionDescriptor* attributes conform to the RDF/XML syntax as defined in RDF 1.1 XML Syntax [Error: Reference source not found].If any of those attributes does not conform, the Hosting CSE shall generate a ***Response Status Code*** indicating a "BAD\_REQUEST" error.

b) The hosting CSE shall also check that the *memberFilter* attribute conforms to a valid SPARQL query request [Error: Reference source not found]. If not, then the Receiver shall reject the Request with a ***Response Status Code*** indicating a "INVALID\_SPARQL\_QUERY" error.

##### 7.4.49.2.2 Retrieve

***Originator:***

No change from the generic procedures in clause Error: Reference source not found.

***Receiver:***

No change from the generic procedures in clause Error: Reference source not found.

##### 7.4.49.2.3 Update

***Originator:***

No change from the generic procedures in clause Error: Reference source not found.

***Receiver:***

No change from the generic procedures in clause 7.2.2.2 with the following exceptions:

1. Primitive specific operation on Recv-6.4 "Check validity of resource representation for the given resource type":

a) If any of those attributes (*inputDescriptor, outputDescriptor* and *functionDescriptor*) is being updated, The Hosting CSE shall check that the new values of those attributes being updated conform to the RDF/XML syntax as defined in RDF 1.1 XML Syntax [Error: Reference source not found]. If any of the new values of those attributes does not conform, the Hosting CSE shall generate a ***Response Status Code*** indicating a "BAD\_REQUEST" error.

b) If the *memberFilter* attribute is being updated, the hosting CSE shall check that the new value of the *memberFilter* attribute conforms to a valid SPARQL query request [Error: Reference source not found]. If not, then the Receiver shall reject the Request with a ***Response Status Code*** indicating a "INVALID\_SPARQL\_QUERY" error.

##### 7.4.49.2.4 Delete

***Originator:***

No change from the generic procedures in clause Error: Reference source not found.

***Receiver:***

No change from the generic procedures in clause 7.2.2.2 with the following exceptions:

1. Primitive specific operation on Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed":

a) The Hosting CSE shall set a NULL value into the *smjpID* attribute of each <semanticMashupInstance> resource that is referenced by this resource's *smiID*.

-------------------------------------------------- End of Change 7---------------------------------------