OCF "Ipanema" – Intended use of Query parameters can be ambiguous – Core Technology WG CR 2807

Legal Disclaimer

THIS IS A DRAFT SPECIFICATION DOCUMENT ONLY AND HAS NOT BEEN ADOPTED BY THE OPEN CONNECTIVITY FOUNDATION. THIS DRAFT DOCUMENT MAY NOT BE RELIED UPON FOR ANY PURPOSE OTHER THAN REVIEW OF THE CURRENT STATE OF THE DEVELOPMENT OF THIS DRAFT DOCUMENT. THE OPEN CONNECTIVITY FOUNDATION AND ITS MEMBERS RESERVE THE RIGHT WITHOUT NOTICE TO YOU TO CHANGE ANY OR ALL PORTIONS HEREOF, DELETE PORTIONS HEREOF, MAKE ADDITIONS HERETO, DISCARD THIS DRAFT DOCUMENT IN ITS ENTIRETY OR OTHERWISE MODIFY THIS DRAFT DOCUMENT AT ANY TIME. YOU SHOULD NOT AND MAY NOT RELY UPON THIS DRAFT DOCUMENT IN ANY WAY, INCLUDING BUT NOT LIMITED TO THE DEVELOPMENT OF ANY PRODUCTS OR SERVICES. IMPLEMENTATION OF THIS DRAFT DOCUMENT IS DONE AT YOUR OWN RISK AMEND AND IT IS NOT SUBJECT TO ANY LICENSING GRANTS OR COMMITMENTS UNDER THE OPEN CONNECTIVITY FOUNDATION INTELLECTUAL PROPERTY RIGHTS POLICY OR OTHERWISE. IN CONSIDERATION OF THE OPEN CONNECTIVITY FOUNDATION GRANTING YOU ACCESS TO THIS DRAFT DOCUMENT, YOU DO HEREBY WAIVE ANY AND ALL CLAIMS ASSOCIATED HEREWITH INCLUDING BUT NOT LIMITED TO THOSE CLAIMS DISCUSSED BELOW, AS WELL AS CLAIMS OF DETRIMENTAL RELIANCE.

***** Change #1 *****

6.2.2 Resource identification and addressing

A Resource may be identified using a URI and addressed by the same URI if the URI is a URL. In some cases, a Resource may need an identifier that is different from a URI; in this case, the Resource may have a Property whose value is the identifier. When the URI is in the form of a URL, then the URI may be used to address the Resource.

An OCF URI is based on the general form of a URI as defined in IETF RFC 3986 as follows (note that the portion in square brackets is optional):

```
<scheme>://<authority>/<path>[?<query>]
```

Specifically, the OCF URI is specified in the following form:

```
ocf://<authority>/<path>[?<query>]
```

The following is a description of values that each component takes.

The "scheme" for the URI is "ocf". The "ocf" scheme represents the semantics, definitions and use as defined in this document. If a URI has the portion preceding the "//" (double slash) omitted, then the "ocf" scheme shall be assumed.

Each transport binding is responsible for specifying how an OCF URI is converted to a transport protocol URI before sending over the network by the requestor. Similarly on the receiver side, each transport binding is responsible for specifying how an OCF URI is converted from a transport protocol URI before handing over to the Resource model layer on the receiver.

The "authority" of an OCF URI shall be the Device ID ("di") value, as defined in [OCF Security], of the Server.

The "path" is a string that unambiguously identifies or references a Resource within the context of the Server. In this version of the document, a path shall not include pct-encoded non-ASCII characters or NUL characters. A *path* shall be preceded by a "/" (slash). The *path* may have "/" (slash) separated segments for human readability reasons. In the OCF context, the "/" (slash) separated segments are treated as a single string that directly references the Resources (i.e. a flat structure) and not parsed as a hierarchy. On the Server, the path or some substring in the path may be shortened by using hashing or some other scheme provided the resulting reference is unique within the context of the host.

Once a path is generated, a Client accessing the Resource or recipient of the URI should use that path as an opaque string and should not parse to infer a structure, organization or semantic.

The "query" is a string that shall contain one or more"<name>=<value>" constructs (aka name-value pair). Where multiple such constructs are supported, each is separated by an "&" (ampersand); this is not a logical "and" operation, but purely a delimiter. Where the use of a query is supported, how the query is handled by the recipient thereof is explicitly defined by the relevant clause in this document or other specifications. The query string will be mapped to the appropriate syntax of the protocol used for messaging. (e.g., CoAP).

***** Change #2 *****

7.6.3.4.2 General requirements for realizations of the batch OCF Interface

All realisations of the batch OCF Interface adhere to the following:

..snip..

– The Client may specify the Resource Type(s) of the linked Resources to which the request is forwarded by including one or more "rt" query parameters in the request, each separated by an "&" as a delimiter (e.g. "?if=oic.if.b&rt=oic.r.switch.binary"). The Server shall then process such additional query parameters in a request that includes "oic.if.b", as selectors for the Linked Resources that are to be processed by the request.

***** Change #3 *****

See changes to the Table below, this is Table 9 – Batch OCF Interface Example

Resources

```
/a/room/1
{
  "rt": "x.org.example.rt.room"],
  "if": ["oic.if.rw","oic.if.baseline","oic.if.b","oic.if.ll"],
  "x.org.example.colour": "blue",
  "x.org.example.dimension": "15bx15wx10h",
  "links": [
     {"href": "/a/room/1", "rel": ["self", "item"], "rt":
["x.org.example.rt.room"], "if":
["oic.if.rw","oic.if.baseline","oic.if.b","oic.if.ll"],"p": {"bm": 2} },
     {"href": "/the/light/1", "rel": ["item"], "rt": ["oic.r.switch.binary"],
"if": ["oic.if.a","oic.if.baseline"], "ins": "11111", "p": {"bm": 2} },
     {"href": "/the/light/2", "rel": ["item"], "rt": ["oic.r.switch.binary"],
"if": ["oic.if.a" ,"oic.if.baseline"], "ins": "22222", "p": {"bm": 2} },
     {"href": "/my/fan/1", "rel": ["item"], "rt": ["oic.r.switch.binary"],
"if": ["oic.if.a", "oic.if.baseline"], "ins": "33333", "p": {"bm": 2} },
     {"href": "/his/fan/2", "rel": ["item"], "rt": ["oic.r.switch.binary"],
"if": ["oic.if.a", "oic.if.baseline"], "ins": "44444", "p": {"bm": 2} },
{"href": "/the/presence/1", "rel": ["item"], "rt": ["oic.r.sensor.presence"],
"if": ["oic.if.s", "oic.if.baseline"], "ins": "55555", "p": {"bm": 2} },
     {"href": "/the/switches/1", "rel": ["item"], "rt": ["oic.wk.col"],
"if":["oic.if.ll", "oic.if.b", "oic.if.baseline"], "ins": "55555", "p": {"bm":
2} }
  ]
}

/the/light/1
{
  "rt": ["oic.r.switch.binary"],
  "if": ["oic.if.a", "oic.if.baseline"],
  "value": false
}

/the/light/2
{
  "rt": ["oic.r.switch.binary"],
  "if": ["oic.if.a", "oic.if.baseline"],
  "value": true
}

/my/fan/1
{
  "rt": ["oic.r.switch.binary"],
  "if": ["oic.if.a", "oic.if.baseline"],
  "value": true
}

/his/fan/2
{
  "rt": ["oic.r.switch.binary"],
  "if": ["oic.if.a", "oic.if.baseline"],
  "value": false
}

/the/presence/1
{
  "rt": ["oic.r.sensor.presence"],
  "if": ["oic.if.s","oic.if.baseline"],
  "value": false
}
```

	/the/switches/1
	{
	  "rt": ["oic.wk.col"],
	  "if":["oic.if.ll", "oic.if.b", "oic.if.baseline"],
	"links": [
	  {
	    "href": "/switch-1a",
	    "rt": ["oic.r.switch.binary"],
	    "if": ["oic.if.a","oic.if.baseline"],
	    "p": {"bm": 2}
	  }
	  {
	    "href": "/switch-1b",
	    "rt": ["oic.r.switch.binary"],
	    "if": ["oic.if.a","oic.if.baseline"],
	    "p": {"bm": 2 }
	   }
	]
	}

Use of batch, successful response	Request: GET /a/room/1?if=oic.if.b
	Becomes the following individual request messages issued by the Device in the Client role
	GET /a/room/1 (NOTE: uses the Default OCF Interface as specified for the Collection Resource, in this example oic.if.rw)
	GET /the/light/1 (NOTE: Uses the Default OCF Interface as specified for this Resource)
	GET /the/light/2 (NOTE: Uses the Default OCF Interface as specified for this Resource)
	GET /my/fan/1 (NOTE: Uses the Default OCF Interface as specified for this Resource)
	GET /his/fan/2 (NOTE: Uses the Default OCF Interface as specified for this Resource)
	GET /the/presence/1 (NOTE: Uses the Default OCF Interface as specified for this Resource)
	GET /the/switches/1 (NOTE: Uses the Default OCF Interface for the Collection that is within the Collection)
	Response:
	[
	 {
	   "href": "/a/room/1",
	   "rep": {"x.org.example.colour": "blue","x.org.example.dimension": "15bx15wx10h"}
	 },
	 {
	   "href": "/the/light/1",
	   "rep": {"value": false}
	 },
	 {
	   "href": "/the/light/2",
	   "rep": {"value": true}
	 },
	 {
	   "href": "/my/fan/1",
	   "rep": {"value": true}
	 },
	 {
	   "href": "/his/fan/2",
	    "rep": {"value": false}
	 },
	 {
	   "href": "/the/presence/1",

```
   "rep": {"value": false}
 },
 {
   "href": "/the/switches/1",
   "rep": [
     {
       "href": "/switch-1a",
       "rt": ["oic.r.switch.binary"],
       "if": ["oic.if.a","oic.if.baseline"],
       "p": {"bm": 2},
       "eps":[
         {"ep": "coaps://[2001:db8:a::b1d4]:55555"}
       ]
     },
     {
       "href": "/switch-1b",
       "rt": ["oic.r.switch.binary"],
       "if": ["oic.if.a","oic.if.baseline"],
       "p": {"bm": 2 },
       "eps":[
         {"ep": "coaps://[2001:db8:a::b1d4]:55555"}
       ]
     }   ]
  }
]
```

Use of batch, error response

Should any of the RETRIEVE requests in the previous example fail then the response includes an empty payload for that Resource instance and an error code is sent. The following example assumes errors from "/my/fan/1" and "/the/switches/1"

Error Response:

```
[
 {
   "href": "/a/room/1",
   "rep": {"x.org.example.colour": "blue","x.org.example.dimension":
"15bx15wx10h"}
 },
 {
   "href": "/the/light/1",
   "rep": {"value": false}
 },
 {
   "href": "/the/light/2",
   "rep": {"value": true}
 },
 {
   "href": "/my/fan/1",
   "rep": {}
 },
 {
   "href": "/his/fan/2",
    "rep": {"value": false}
 },
{
   "href": "/the/presence/1",
   "rep": {"value": false}
 },
 {
   "href": "/the/switches/1",
   "rep": {}
 }
```

	]

Use of batch
(UPDATE has POST semantics)
	UPDATE /a/room/1?if=oic.if.b
	[
	  {
	    "href": "/the/light/1",
	    "rep": {
	      "value": false
	    }
	  },
	  {
	    "href": "/the/light/2",
	    "rep": {
	      "value": true
	    }
	  },
	  {
	    "href": "/a/room/1",
	    "rep": {
	      "x.org.example.colour": "red"
	    }
	  }
	]
	
	This turns /the/light/1 off, turns /the/light/2 on, and sets the colour of /a/room/1 to "red".
	
	The response will be same as response for GET /a/room/1?if=oic.if.b with the updated Property values as shown.
	[
	 {
	   "href": "/a/room/1",
	   "rep":{"x.org.example.colour": "red",
	     "x.org.example.dimension": "15bx15wx10h"}
	 },
	 {
	   "href": "/the/light/1",
	   "rep": {"value": false}
	 },
	 {
	   "href": "/the/light/2",
	   "rep": {"value": true}
	 }
	]
	Example use of additional query parameters to select items by matching Link Parameters.
	
	Retrieving all items that are Presence Sensors ("oic.r.sensor.presence"):
	RETRIEVE /a/room/1?if=oic.if.b&rt=oic.r.sensor.presence
	
	Response payload:
	[
	  {
	    "href": "/the/presence/1",
	    "rep": {
	      "value": false
	    }
	  }
	]

***** Change #4 *****

7.6.3.5 Actuator OCF Interface

The actuator OCF Interface is the OCF Interface for viewing Resources that may be actuated i.e. changes some value within or the state of the entity abstracted by the Resource:

- The actuator OCF Interface name shall be "oic.if.a"
- The actuator OCF Interface shall expose in the Resource Representation all mandatory Properties as defined by the applicable OpenAPI 2.0 schema; the actuator OCF Interface may also expose in the Resource Representation optional Properties as defined by the applicable OpenAPI 2.0 schema that are implemented by the target Device.

For example, a "Heater" Resource (for illustration only):

```
/a/act/heater
{
   "rt": ["x.com.acme.gas"],
   "if": ["oic.if.baseline", "oic.if.r", "oic.if.a", "oic.if.s"],
   "x.com.acme.settemp": 10,
   "x.com.acme.currenttemp" : 7
}
```

The actuator OCF Interface with respect to "Heater" Resource (for illustration only):

a) Retrieving values of an actuator.

```
Request: RETRIEVE /a/act/heater?if="oic.if.a"

Response: Content
Payload:
{
  "x.com.acme.settemp": 10,
  "x.com.acme.currenttemp" : 7
}
```

b) Correct use of actuator OCF Interface.

```
Request: UPDATE /a/act/heater?if="oic.if.a"
{
  "x.com.acme.settemp": 20
}
Response: Changed
Payload:
{
  "x.com.acme.settemp": 20
}
```

c) Incorrect use of actuator OCF Interface.

```
Request: UPDATE /a/act/heater?if="oic.if.a"
{
  "if": ["oic.if.s"]   ← this is visible through baseline OCF Interface
}
Response:Bad Request
Payload:
{
}
```

- A RETRIEVE request using this OCF Interface shall return the Representation for this Resource as defined by the applicable OpenAPI 2.0 schema, subject to any query parameters that may also be defined as part of the applicable OpenAPI 2.0 schema.
- An UPDATE request using this OCF Interface shall provide a payload or body that contains the Properties that will be updated on the target Resource.

7.6.3.6 Sensor OCF Interface

The sensor OCF Interface is the OCF Interface for retrieving measured, sensed or capability specific information from a Resource that senses:

- The sensor OCF Interface name shall be "oic.if.s".
- The sensor OCF Interface shall expose in the Resource Representation all mandatory Properties as defined by the applicable OpenAPI 2.0 schema; the sensor OCF Interface may also expose in the Resource Representation optional Properties as defined by the applicable OpenAPI 2.0 schema that are implemented by the target Device.
- A RETRIEVE request using this OCF Interface shall return this representation for the Resource as defined by the applicable OpenAPI 2.0 schema, subject to any query parameters that may also be defined as part of the applicable OpenAPI 2.0 schema.

***** Change #5 *****

7.9 Query Parameters

7.9.1 Introduction

A query string is a fundamental part of the definition of a URI (see 6.2.2). The definition of a query may include Properties and Link Parameters by declaring the Property or Link Parameter (i.e. <Property name, Link Parameter name> = <desired Property value, Link Parameter value>) as one of the segments of the query. Only ASCII strings are permitted in queries, and NULL characters are disallowed in queries. This means that only Property and Link Parameter values with ASCII characters may be matched in a query.

When a query is defined as a selector, a Resource is selected when all the declared Properties or Link Parameters in the query match the corresponding Properties or Link Parameters in the target..

The processing of any query parameter by a Server is as specified in this document or other OCF specifications. For any query parameters that are not explicitly specified, the Server may ignore those query parameters and the request is processed as if the query parameter did not exist in the request.

7.9.2 Use of multiple parameters within a query

When a query contains multiple separate query parameters these are delimited by an "&" as described in 6.2.2. Multiple query parameters are only applicable to Collections or Resources with a multi-value "rt".

A Client may select a specific Resource type using separate query parameters, for example "?if=oic.if.b&rt=oic.r.switch.binary". If such queries are supported by the Server this shall be accomplished by matching "all of" the different query parameter types received (i.e. "rt", "if") against the target of the query. In the example, this resolves to a batch response that includes only instances of oic.r.switch.binary. There is no significance applied to the order of the query parameters.

A Client may select more than one Resource Type using repeated query parameters, for example "?rt=oic.r.switch.binary&rt=oic.r.ramptime". If such queries are supported by the Server, this shall be accomplished by matching "any of" the repeated query parameters against the target of the query. In the example, any instances of "oic.r.switch.binary" and/or "oic.r.ramptime" that may exist are selected.

A Client may select multiple Resource Types using multiple repeated "rt" parameters in addition to a separate "if" parameter in a single query, for example "?if=oic.if.b&rt=oic.r.switch.binary&rt=oic.r.ramptime". If such queries are supported by the Server, this shall be accomplished by matching "any of" the repeated query parameters and then matching "all of" the different query parameter types. In the example any instances of "oic.r.switch.binary" and/or "oic.r.ramptime" that may exist are selected in a batch response.

NOTE The parameters within a query string are represented within the actual messaging protocol as defined in clause 12.2.2

7.9.3 Application to multi-value "rt" Resources

An "rt" query for a multi-value "rt" Resource with the Default OCF Interface of "oic.if.a", "oic.if.s", "oic.if.r", "oic.if.rw" or "oic.if.baseline" is an extension of a generic "rt" query.

When a Server receives a RETRIEVE request for a multi-value "rt" Resource with an "rt" query, (e.g.. GET /ResExample?rt=oic.r.foo), the Server should respond only when the query value is an item of the "rt" Property Value of the target Resource and should send back only the Properties associated with the query value(s). For example, upon receiving GET /ResExample?rt=oic.r.switch.binary targeting a Resource with "rt": ["oic.r.switch.binary", "oic.r.light.brightness"], the Server responds with only the Properties of "oic.r.switch.binary".

When a Server receives an UPDATE request for a multi-value "rt" Resource with an "rt" query, (e.g.POST /ResExample?rt=oic.r.foo), the Server should only apply the payload received to the Properties that are part of the "oic.r.foo" Resource.

7.9.4 OCF Interface specific considerations for queries

7.9.4.1 OCF Interface selection

When an OCF Interface is to be selected for a request, it shall be specified as a query parameter in the URI of the Resource in the request message. If no query parameter is specified, then the Default OCF Interface shall be used. If the selected OCF Interface is not one of the permitted OCF Interfaces on the Resource, then selecting that OCF Interface is an error and the Server shall respond with an error response code. A Client shall not include more than one OCF Interface in a query parameter. If a Server receives a request that has more than one OCF Interface included in a query parameter (e.g. "?if=oic.if.ll&if=oic.if.rw") then the Server may either reject the request with an appropriate non-success path response, or the Server may attempt to process the request using the first "if" received

For example, the baseline OCF Interface may be selected by adding "if=oic.if.baseline" to the list of query parameters in the URI of the target Resource. For example: "GET /oic/res?if=oic.if.baseline".

7.9.4.2 Batch OCF Interface

See 7.6.3.4 for details on the batch OCF Interface itself. Query parameters may be used with the batch OCF Interface in order to select particular Resources in a Collection for retrieval or update; these parameters are used to select items in the Collection by matching Link Parameter Values.

When Link selection query parameters are used with RETRIEVE operations applied using the batch OCF Interface, only the Resources in the Collection with matching Link Parameters should be returned.

When Link selection query parameters are used with UPDATE operations applied using the batch OCF Interface, only the Resources having matching Link Parameters should be updated.

See 7.6.3.4.5 for examples of RETRIEVE and UPDATE operations that use Link selection query parameters.