



OCF 2.3 – U+ Resource Mapping specification - BTG

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.

The OCF logo is a trademark of Open Connectivity Foundation, Inc. in the United States or other countries. *Other names and brands may be claimed as the property of others.

Copyright © 2018 Open Connectivity Foundation, Inc. All rights reserved.

Copying or other form of reproduction and/or distribution of these works are strictly prohibited.



40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81

CONTENTS

1	Scope	3
2	Normative references	3
3	Terms, definitions symbols and abbreviations	3
3.1	Terms and definitions	3
3.2	Symbols and abbreviations	4
3.3	Conventions	4
4	Document conventions and organization	4
4.1	Notation	4
4.2	Data types	5
5	Theory of Operation	5
5.1	Interworking Approach	5
5.2	Mapping Syntax	5
5.2.1	General	5
5.2.2	Value Assignment	5
5.2.3	Property Naming	5
5.2.4	Arrays	5
5.2.5	Default Mapping	5
5.2.6	Conditional Mapping	5
5.2.7	Loops	6
5.2.8	Method Invocation	6
6	Device Type Mapping	6
6.1	Introduction	6
6.2	U+ Device Types to OCF Device Types	6
7	Resource to U+ Property Equivalence	6
7.1	Introduction	6
7.2	U+ Property to OCF Resources	7
8	Detailed Mapping APIs	7
8.1	Air Conditioner Mapping	7
8.1.1	Derived Model	7
8.1.2	Property Definition	7
8.1.3	Derived Model Definition	8
8.2	Air Purifier Mapping	9
8.2.1	Derived Model	9
8.2.2	Property Definition	9
8.2.3	Derived Model Definition	9
8.3	Water Heater Mapping	10
8.3.1	Derived Model	10
8.3.2	Property Definition	10



82	8.3.3	Derived Model Definition	11
----	-------	--------------------------------	----

83

84 Tables

85

86	Table 1 U+ to OCF Device Type Mapping	6
----	---	---

87	Table 2 U+ Property to OCF Resource Type Mapping	7
----	--	---

88	Table 3 The property mapping for uplus.device.airconditioner	7
----	--	---

89	Table 4 The properties of uplus.device.airconditioner	8
----	---	---

90	Table 5 The property mapping for uplus.device.airpurifier.....	9
----	--	---

91	Table 6 The properties of uplus.device.airpurifier	9
----	--	---

92	Table 7 The property mapping for uplus.device.waterheater	10
----	---	----

93	Table 8 The properties of uplus.device.waterheater.....	11
----	---	----

94

95 1 Scope

96 The OCF Resource to U+ Property Mapping specification (“this specification”) provides detailed
97 mapping information between OCF Resources and U+ Properties.

98 2 Normative references

99 The following documents, in whole or in part, are normatively referenced in this document and are
100 indispensable for its application. For dated references, only the edition cited applies. For undated
101 references, the latest edition of the referenced document (including any amendments) applies.

102 OCF Core Specification

103 OCF Device Specification

104 OCF Resource Type Specification

105 OCF Security Specification

106 OCF Bridging Specification

107 Derived Models for Interoperability between IoT Ecosystems, Stevens & Merriam, March 2016
108 [https://www.iab.org/wp-content/IAB-uploads/2016/03/OCF-Derived-Models-for-Interoperability-](https://www.iab.org/wp-content/IAB-uploads/2016/03/OCF-Derived-Models-for-Interoperability-Between-IoT-Ecosystems_v2-examples.pdf)
109 [Between-IoT-Ecosystems_v2-examples.pdf](https://www.iab.org/wp-content/IAB-uploads/2016/03/OCF-Derived-Models-for-Interoperability-Between-IoT-Ecosystems_v2-examples.pdf)

110 Haier Household Air Conditioner Device Model v1.8

111 Haier Air Cube Device Model v0.1

112 Electric Water Heater Device Model v1.3

113 3 Terms, definitions symbols and abbreviations

114 All terms and definitions as defined in the OCF Core Specification, OCF Security Specification
115 , and OCF Bridging Specification also apply to this specification.

116 3.1 Terms and definitions

117 None defined.



118 **3.2 Symbols and abbreviations**

119 None defined.

120 **3.3 Conventions**

121 In this specification a number of terms, conditions, mechanisms, sequences, parameters, events,
122 states, or similar terms are printed with the first letter of each word in uppercase and the rest
123 lowercase (e.g., Network Architecture). Any lowercase uses of these words have the normal
124 technical English meaning.

125 **4 Document conventions and organization**

126 For the purposes of this document, the terms and definitions given in OCF Core Specification and
127 OCF Security Specification

128 apply.

129 **4.1 Notation**

130 In this document, features are described as required, recommended, allowed or DEPRECATED as
131 follows:

132 Required (or shall or mandatory).

133 These basic features shall be implemented to comply with the Mapping Specification. The
134 phrases “shall not”, and “PROHIBITED” indicate behavior that is prohibited, i.e. that if
135 performed means the implementation is not in compliance.

136 Recommended (or should).

137 These features add functionality supported by the Mapping Specification and should be
138 implemented. Recommended features take advantage of the capabilities the Mapping
139 Specification, usually without imposing major increase of complexity. Notice that for compliance
140 testing, if a recommended feature is implemented, it shall meet the specified requirements to
141 be in compliance with these guidelines. Some recommended features could become
142 requirements in the future. The phrase “should not” indicates behavior that is permitted but not
143 recommended.

144 Allowed (or allowed).

145 These features are neither required nor recommended by the Mapping Specification, but if the
146 feature is implemented, it shall meet the specified requirements to be in compliance with these
147 guidelines.

148 Conditionally allowed (CA)

149 The definition or behaviour depends on a condition. If the specified condition is met, then the
150 definition or behaviour is allowed, otherwise it is not allowed.

151 Conditionally required (CR)

152 The definition or behaviour depends on a condition. If the specified condition is met, then the
153 definition or behaviour is required. Otherwise the definition or behaviour is allowed as default
154 unless specifically defined as not allowed.

155 DEPRECATED

156 Although these features are still described in this specification, they should not be implemented
157 except for backward compatibility. The occurrence of a deprecated feature during operation of
158 an implementation compliant with the current specification has no effect on the
159 implementation’s operation and does not produce any error conditions. Backward compatibility



160 may require that a feature is implemented and functions as specified but it shall never be used
161 by implementations compliant with this specification.

162 Strings that are to be taken literally are enclosed in “double quotes”.

163 Words that are emphasized are printed in *italic*.

164 **4.2 Data types**

165 See OCF Core Specification.

166 **5 Theory of Operation**

167 **5.1 Interworking Approach**

168 The interworking between U+ Properties and OCF Resource Types is modelled using the derived
169 model syntax described in Derived Models for Interoperability.

170 **5.2 Mapping Syntax**

171 Within the defined syntax for derived modelling used by this Specification there are two blocks that
172 define the actual Property-Property equivalence or mapping. These blocks are identified by the
173 keywords ‘x-to-ocf’ and ‘x-from-ocf’. Derived Models for Interoperability does not define a rigid
174 syntax for these blocks; they are free form string arrays that contain pseudo-coded mapping logic.
175 Within this specification we apply the rules in the following sub-sections to these blocks to ensure
176 consistency and re-usability and extensibility of the mapping logic that is defined.

177 **5.2.1 General**

178 All statements are terminated with a carriage return.

179 **5.2.2 Value Assignment**

180 The equals sign (=) is used to assign one value to another. The assignee is on the left of the
181 operator; the value being assigned on the right.

182 **5.2.3 Property Naming**

183 All Property names are identical to the name used by the original model; for example, from the
184 OCF Temperature Resource the Property name ‘temperature’ is used whereas when referred to
185 the derived ecosystem then the semantically equivalent Property name is used.

186 When the same name is used by both OCF and the derived ecosystem for semantically equivalent
187 values then the name of the OCF Property is prepended by the ecosystem designator ‘ocf’ to avoid
188 ambiguity (e.g. ‘ocf.step’)

189 **5.2.4 Arrays**

190 An array element is indicated by the use of square brackets ‘[]’ with the index of the element
191 contained therein, e.g. range[1]. All arrays start at an index of 0. If an entire array is being
192 referenced then no index is included, e.g. selectablehumiditylevels[].

193 **5.2.5 Default Mapping**

194 There are cases where the specified mapping is not possible as one or more of the Properties
195 being mapped are optional in the source model. In all such instances a default mapping is provided.
196 The default map is indicated by the prepending of an ‘otherwise:’ modifier to the assignment. (e.g.
197 ‘otherwise: step = 1’)

198 **5.2.6 Conditional Mapping**

199 When a mapping is dependent on the meeting of other conditions then the syntax:

200 if ‘condition’, ‘mapping’.

201 Is applied.



202 E.g. if step >0, ocf.step = step.

203 5.2.7 Loops

204 When a mapping can be represented by a repeated loop governed by some condition then the
205 syntax:

206 for 'initialize', 'condition', 'increment': 'mapping'

207 Where:

208 'initialize' is an initial local loop control variable setting.

209 'condition' is the loop controller, the loop repeats until the condition evaluates to 'false'.

210 'increment' allows for update of the control variable, if omitted an increment of '1' is assumed.

211 Is applied.

212 E.g. for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
213 modearray[supportedmodes[x]]

214 5.2.8 Method Invocation

215 The invocation of a method or remote procedure call (RPC) from the derived ecosystem as part of
216 the mapping from an OCF Resource is indicated by the use of a double colon '::' delimiter between
217 the applicable resource, service, interface or other construct identifier and the method or RPC
218 name. The method name always includes trailing parentheses which would include any parameters
219 should they be passed.

220 For example, when dealing with the switchon() method from AllJoyn this gives a complete method
221 invocation as: operation.oncontrol::switchon().

222 6 Device Type Mapping

223 6.1 Introduction

224 This Section contains the mappings from U+ Device Types to OCF Device Types.

225 6.2 U+ Device Types to OCF Device Types

226 Table 1 captures the equivalency mapping between U+ Device Types and OCF Device Types.

227 **Table 1 U+ to OCF Device Type Mapping**

U+ Device Type	OCF Device Type
Air Conditioner	oic.d.airconditioner
Air Purifier	oic.d.airpurifier
Water Heater	oic.d.waterheater

228 7 Resource to U+ Property Equivalence

229 7.1 Introduction

230 This Section lists the U+ Properties and provides the equivalent OCF Resource Type(s) to which
231 the Properties map.



232 **7.2 U+ Property to OCF Resources**

233 Table 2 captures the equivalency mapping between U+ Properties and OCF Resource Types. U+
 234 properties are device type-dependent. The properties with same name may be different within
 235 different device types.

236 **Table 2 U+ Property to OCF Resource Type Mapping**

U+ Device Type	U+ Property	OCF Resource Type Name	OCF Resource Type ID	OCF Interface(s)
Air Conditioner	onOffStatus	Binary Switch	oic.r.switch.binary	oic.if.a
	targetTemperature	Temperature	oic.r.temperature	oic.if.s oic.if.a
	windSpeed	Selectable Levels	oic.r.selectablelevels	oic.if.a
	operationMode	Mode	oic.r.mode	oic.if.a
Air Purifier	onOffStatus	Binary Switch	oic.r.switch.binary	oic.if.a
	windSpeed	Selectable Levels	oic.r.selectablelevels	oic.if.a
	operationMode	Mode	oic.r.mode	oic.if.a
Water Heater	onOffStatus	Binary Switch	oic.r.switch.binary	oic.if.a
	targetTemperature	Temperature	oic.r.temperature	oic.if.s oic.if.a

237 **8 Detailed Mapping APIs**

238 This section provides a device type mapping description (using JSON that aligns with the Derived
 239 Modelling syntax described in [Derived Model White Paper]) for all U+ Properties and OCF
 240 Resources that are within scope.

241 **8.1 Air Conditioner Mapping**

242 **8.1.1 Derived Model**

243 The derived model: `uplus.device.airconditioner`.

244 **8.1.2 Property Definition**

245 **Table 3 The property mapping for `uplus.device.airconditioner`**

Uplus Property name	OCF Resource	To OCF	From OCF
targetTemperature	oic.r.temperature	oic.r.temperature.temperature=targetTemperature	targetTemperature=oic.r.temperature.temperature



operationMode	oic.r.mode	supportedModes=["Auto","Cool","Dry","Warm","Wind"]modes=supportedModes[operationMode]	operationMode=supportedModes.indexof(modes)
onOffStatus	oic.r.switch.binary.value	oic.r.switch.value = onOffStatus	onOffStatus=oic.r.switch.value
windSpeed	oic.r.selectablelevels	availablelevels=[1,2,3,4,5]targetlevel=windSpeed	windSpeed=targetlevel

246

Table 4 The properties of uplus.device.airconditioner

Uplus name	Property	Type	Required	Description
targetTemperature		number	yes	target temperature
operationMode		integer	yes	
onOffStatus		boolean	yes	the switch of air conditioner
windSpeed		integer	yes	wind speed

247

8.1.3 Derived Model Definition

248

```

249 {
250   "id": "http://openinterconnect.org/uplusocfmapping/schemas/uplus.device.airconditioner.json#",
251   "$schema": "http://json-schema.org/draft-04/schema#",
252   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
253   "title": "AirConditioner Mapping",
254   "definitions": {
255     "uplus.device.airconditioner": {
256       "type": "object",
257       "properties": {
258         "onOffStatus": {
259           "type": "boolean",
260           "description": "the switch of air conditioner",
261           "x-ocf-conversion": {
262             "x-ocf-alias": "oic.r.switch.binary.value",
263             "x-to-ocf": [
264               "oic.r.switch.binary.value = onOffStatus"
265             ],
266             "x-from-ocf": [
267               "onOffStatus=oic.r.switch.binary.value"
268             ]
269           }
270         },
271         "targetTemperature": {
272           "type": "number",
273           "description": "target temperature",
274           "x-ocf-conversion": {
275             "x-ocf-alias": "oic.r.temperature.temperature",
276             "x-to-ocf": [
277               "oic.r.temperature.temperature=targetTemperature"
278             ],
279             "x-from-ocf": [
280               "targetTemperature=oic.r.temperature.temperature"
281             ]
282           }
283         },
284         "windSpeed": {
285           "type": "integer",
286           "description": "wind speed",
287           "x-ocf-conversion": {
288             "x-ocf-alias": "oic.r.selectablelevels",
289             "x-to-ocf": [
290               "availablelevels=[1,2,3,4,5]",
291               "targetlevel=windSpeed"
292             ],
293             "x-from-ocf": [
294               "windSpeed=targetlevel"

```




```

294     ]
295   }
296 },
297   "operationMode": {
298     "type": "integer" ,
299     "description": "",
300     "x-ocf-conversion": {
301       "x-ocf-alias": "oic.r.mode",
302       "x-to-ocf": [
303         "supportedModes= [\"Auto\", \"Cool\", \"Dry\", \"Warm\", \"Wind\"]",
304         "modes=supportedModes[operationMode]"
305       ],
306       "x-from-ocf": [
307         "operationMode= supportedModes.indexOf(modes)"
308       ]
309     }
310   }
311 }
312 }
313 },
314 "type": "object",
315 "allOf": [
316   {"$ref": "#/definitions/uplus.device.airconditioner"}
317 ],
318 "required": ["onOffStatus", "targetTemperature", "windSpeed", "operationMode"]
319 }

```

320 8.2 Air Purifier Mapping

321 8.2.1 Derived Model

322 The derived model: uplus.device.airpurifier.

323 8.2.2 Property Definition

324 **Table 5 The property mapping for uplus.device.airpurifier**

Uplus Property name	OCF Resource	To OCF	From OCF
operationMode	oic.r.mode	supportedModes=["Auto","Quiet","Sleep"]modes=supportedModes[operationMode]	operationMode=supportedModes.indexOf(modes)
onOffStatus	oic.r.switch.binary.value	oic.r.switch.value = onOffStatus	onOffStatus = oic.r.switch.value
windSpeed	oic.r.selectable levels	availablelevels=[0,1,2,3,4]targetlevel=windSpeed	windSpeed=targetlevel

325 **Table 6 The properties of uplus.device.airpurifier**

Uplus Property name	Type	Required	Description
operationMode	integer	yes	
onOffStatus	boolean	yes	the switch of air purifier
windSpeed	integer	yes	wind speed

326 8.2.3 Derived Model Definition

```

327 {
328   "id": "http://openinterconnect.org/uplusocfmapping/schemas/uplus.device.airpurifier.json#",
329   "$schema": "http://json-schema.org/draft-04/schema#",
330   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
331   "title": "AirPurifier Mapping",
332   "definitions": {
333     "uplus.device.airpurifier": {
334       "type": "object",
335       "properties": {

```



```

336     "onOffStatus": {
337         "type" : "boolean",
338         "description": "the switch of air purifier",
339         "x-ocf-conversion": {
340             "x-ocf-alias": "oic.r.switch.binary.value",
341             "x-to-ocf": [
342                 "oic.r.switch.binary.value = onOffStatus"
343             ],
344             "x-from-ocf": [
345                 "onOffStatus = oic.r.switch.binary.value"
346             ]
347         }
348     },
349     "windSpeed": {
350         "type": "integer" ,
351
352         "description": "wind speed",
353         "x-ocf-conversion": {
354             "x-ocf-alias": "oic.r.selectablelevels",
355             "x-to-ocf": [
356                 "availablelevels=[0,1,2,3,4]",
357                 "targetlevel=windSpeed"
358             ],
359             "x-from-ocf": [
360                 "windSpeed=targetlevel"
361             ]
362         }
363     },
364     "operationMode": {
365         "type": "integer" ,
366         "description": "",
367         "x-ocf-conversion": {
368             "x-ocf-alias": "oic.r.mode",
369             "x-to-ocf": [
370                 "supportedModes= [\"Auto\", \"Quiet\", \"Sleep\"]",
371                 "modes=supportedModes[operationMode]"
372             ],
373             "x-from-ocf": [
374                 "operationMode= supportedModes.indexOf(modes)"
375             ]
376         }
377     }
378 }
379 }
380 },
381 "type": "object",
382 "allOf": [
383     {"$ref": "#/definitions/uplus.device.airpurifier"}
384 ],
385 "required": ["onOffStatus", "windSpeed", "operationMode"]
386 }

```

387 8.3 Water Heater Mapping

388 8.3.1 Derived Model

389 The derived model: uplus.device.waterheater.

390 8.3.2 Property Definition

391 **Table 7 The property mapping for uplus.device.waterheater**

Uplus Property name	OCF Resource	To OCF	From OCF
targetTemperature	oic.r.temperature.temperature	oic.r.temperature.temperature=targetTemperature	targetTemperature=oic.r.temperature.temperature
onOffStatus	oic.r.switch.binary.value	oic.r.switch.binary.value = onOffStatus	onOffStatus = oic.r.switch.binary.value



392

Table 8 The properties of uplus.device.waterheater

Uplus name	Property	Type	Required	Description
targetTemperature		number	yes	target temperature
onOffStatus		boolean	yes	the switch of water heater

393 **8.3.3 Derived Model Definition**

```

394 {
395   "id": "http://openinterconnect.org/uplusocfmapping/schemas/uplus.device.waterheater.json#",
396   "$schema": "http://json-schema.org/draft-04/schema#",
397   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
398   "title": "WaterHeater Mapping",
399   "definitions": {
400     "uplus.device.waterheater": {
401       "type": "object",
402       "properties": {
403         "onOffStatus": {
404           "type": "boolean",
405           "description": "the switch of water heater",
406           "x-ocf-conversion": {
407             "x-ocf-alias": "oic.r.switch.binary.value",
408             "x-to-ocf": [
409               "oic.r.switch.binary.value = onOffStatus"
410             ],
411             "x-from-ocf": [
412               "onOffStatus = oic.r.switch.binary.value"
413             ]
414           }
415         },
416         "targetTemperature": {
417           "type": "number",
418           "description": "target temperature",
419           "x-ocf-conversion": {
420             "x-ocf-alias": "oic.r.temperature.temperature",
421             "x-to-ocf": [
422               "oic.r.temperature.temperature=targetTemperature"
423             ],
424             "x-from-ocf": [
425               "targetTemperature=oic.r.temperature.temperature"
426             ]
427           }
428         }
429       }
430     }
431   },
432   "type": "object",
433   "allOf": [
434     {"$ref": "#/definitions/uplus.device.warterheater"}
435   ],
436   "required": ["onOffStatus", "targetTemperature"]
437 }
438

```