

OCF Resource to AllJoyn Interface Mapping

VERSION 1.0.0 | June 2017



OPEN CONNECTIVITY
FOUNDATION™

CONTACT admin@openconnectivity.org

Copyright Open Connectivity Foundation, Inc. © 2017.
All Rights Reserved.

Legal Disclaimer

3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

NOTHING CONTAINED IN THIS DOCUMENT SHALL BE DEEMED AS GRANTING YOU ANY KIND OF LICENSE IN ITS CONTENT, EITHER EXPRESSLY OR IMPLIEDLY, OR TO ANY INTELLECTUAL PROPERTY OWNED OR CONTROLLED BY ANY OF THE AUTHORS OR DEVELOPERS OF THIS DOCUMENT. THE INFORMATION CONTAINED HEREIN IS PROVIDED ON AN "AS IS" BASIS, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE AUTHORS AND DEVELOPERS OF THIS SPECIFICATION HEREBY DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. OPEN CONNECTIVITY FOUNDATION, INC. FURTHER DISCLAIMS ANY AND ALL WARRANTIES OF NON-INFRINGEMENT, ACCURACY OR LACK OF VIRUSES.

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 © 2017 Open Connectivity Foundation, Inc. All rights reserved.

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

21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62

CONTENTS

1	Scope	10
2	Normative references	10
3	Terms, definitions symbols and abbreviations	10
3.1	Terms and definitions	10
3.2	Symbols and abbreviations	11
4	Document conventions and organization	11
4.1	Conventions	11
4.2	Notation.....	11
4.3	Data types	12
5	Theory of Operation	12
5.1	Interworking Approach.....	12
5.2	Mapping Syntax.....	12
5.2.1	Block Identification	12
5.2.2	Use of Carriage Return	12
5.2.3	Value Assignment	12
5.2.4	Property Naming	12
5.2.5	Arrays	12
5.2.6	Default Mapping	12
5.2.7	Conditional Mapping.....	13
5.2.8	Loops	13
5.2.9	Method Invocation	13
6	Device Type Mapping.....	13
6.1	Introduction	13
6.2	AllJoyn Device Types to OCF Device Types	13
6.3	OCF Device Types with no AllJoyn Equivalent.....	15
7	Resource to Interface Equivalence	16
7.1	Introduction	16
7.2	AllJoyn Interfaces to OCF Resources	16
8	Detailed Mapping APIs	18
8.1	Introduction	18
8.2	Air Quality Mapping	19
8.2.1	Introduction	19
8.2.2	Example URI	20
8.2.3	Resource Type	20
8.2.4	RAML Definition	20
8.2.5	Property Definition	22
8.2.6	CRUDN behavior	22
8.3	Air Quality Level Mapping.....	22
8.3.1	Introduction	22

63	8.3.2	Example URI	22
64	8.3.3	Resource Type	22
65	8.3.4	RAML Definition	22
66	8.3.5	Property Definition	24
67	8.3.6	CRUDN behavior	25
68	8.4	Current Humidity Mapping	25
69	8.4.1	Introduction	25
70	8.4.2	Example URI	25
71	8.4.3	Resource Type	25
72	8.4.4	RAML Definition	25
73	8.4.5	Property Definition	27
74	8.4.6	CRUDN behavior	27
75	8.5	Current Temperature Mapping	27
76	8.5.1	Introduction	27
77	8.5.2	Example URI	27
78	8.5.3	Resource Type	27
79	8.5.4	RAML Definition	27
80	8.5.5	Property Definition	29
81	8.5.6	CRUDN behavior	29
82	8.6	Target Humidity Mapping	29
83	8.6.1	Introduction	29
84	8.6.2	Example URI	29
85	8.6.3	Resource Type	29
86	8.6.4	RAML Definition	29
87	8.6.5	Property Definition	34
88	8.6.6	CRUDN behavior	35
89	8.7	Target Temperature Mapping	35
90	8.7.1	Introduction	35
91	8.7.2	Example URI	35
92	8.7.3	Resource Type	35
93	8.7.4	RAML Definition	35
94	8.7.5	Property Definition	40
95	8.7.6	CRUDN behavior	40
96	8.8	Audio Volume Mapping	40
97	8.8.1	Introduction	40
98	8.8.2	Example URI	40
99	8.8.3	Resource Type	40
100	8.8.4	RAML Definition	40
101	8.8.5	Property Definition	44
102	8.8.6	CRUDN behavior	44
103	8.9	Climate Control Mode Mapping	44
104	8.9.1	Introduction	44
105	8.9.2	Example URI	44
106	8.9.3	Resource Type	44

107	8.9.4	RAML Definition	44
108	8.9.5	Property Definition	48
109	8.9.6	CRUDN behavior	49
110	8.10	Closed Status Mapping	49
111	8.10.1	Introduction	49
112	8.10.2	Example URI	49
113	8.10.3	Resource Type	49
114	8.10.4	RAML Definition	49
115	8.10.5	Property Definition	50
116	8.10.6	CRUDN behavior	50
117	8.11	Cycle Control Mapping	50
118	8.11.1	Introduction	50
119	8.11.2	Example URI	50
120	8.11.3	Resource Type	50
121	8.11.4	RAML Definition	50
122	8.11.5	Property Definition	52
123	8.11.6	CRUDN behavior	52
124	8.12	Fan Speed Level Mapping	52
125	8.12.1	Introduction	52
126	8.12.2	Example URI	52
127	8.12.3	Resource Type	53
128	8.12.4	RAML Definition	53
129	8.12.5	Property Definition	56
130	8.12.6	CRUDN behavior	56
131	8.13	Heating Zone Mapping	56
132	8.13.1	Introduction	56
133	8.13.2	Example URI	57
134	8.13.3	Resource Type	57
135	8.13.4	RAML Definition	57
136	8.13.5	Property Definition	58
137	8.13.6	CRUDN behavior	59
138	8.14	HVAC Fan Mode Mapping	59
139	8.14.1	Introduction	59
140	8.14.2	Example URI	59
141	8.14.3	Resource Type	59
142	8.14.4	RAML Definition	59
143	8.14.5	Property Definition	62
144	8.14.6	CRUDN behavior	62
145	8.15	On Off Mapping	63
146	8.15.1	Introduction	63
147	8.15.2	Example URI	63
148	8.15.3	Resource Type	63
149	8.15.4	RAML Definition	63
150	8.15.5	Property Definition	67

151	8.15.6	CRUDN behavior	67
152	8.16	Oven Cycle Phase Mapping	67
153	8.16.1	Introduction	67
154	8.16.2	Example URI	67
155	8.16.3	Resource Type	67
156	8.16.4	RAML Definition	67
157	8.16.5	Property Definition	69
158	8.16.6	CRUDN behavior	69
159		Annex A Swagger2.0 (Informative)	70
160	A.1	Audio Volume Mapping	70
161	A.1.1	Introduction	70
162	A.1.2	Example URI	70
163	A.1.3	Resource Type	70
164	A.1.4	Swagger2.0 Definition	70
165	A.1.5	Property Definition	72
166	A.1.6	CRUDN behavior	73
167	A.2	Climate Control Mode Mapping	73
168	A.2.1	Introduction	73
169	A.2.2	Example URI	73
170	A.2.3	Resource Type	73
171	A.2.4	Swagger2.0 Definition	73
172	A.2.5	Property Definition	76
173	A.2.6	CRUDN behavior	76
174	A.3	Closed Status Mapping	77
175	A.3.1	Introduction	77
176	A.3.2	Example URI	77
177	A.3.3	Resource Type	77
178	A.3.4	Swagger2.0 Definition	77
179	A.3.5	Property Definition	78
180	A.3.6	CRUDN behavior	78
181	A.4	Air Quality Mapping	78
182	A.4.1	Introduction	78
183	A.4.2	Example URI	79
184	A.4.3	Resource Type	79
185	A.4.4	Swagger2.0 Definition	79
186	A.4.5	Property Definition	81
187	A.4.6	CRUDN behavior	81
188	A.5	Air Quality Level Mapping	81
189	A.5.1	Introduction	81
190	A.5.2	Example URI	82
191	A.5.3	Resource Type	82
192	A.5.4	Swagger2.0 Definition	82
193	A.5.5	Property Definition	84
194	A.5.6	CRUDN behavior	85

195	A.6	Current Humidity Mapping	85
196	A.6.1	Introduction	85
197	A.6.2	Example URI	85
198	A.6.3	Resource Type	85
199	A.6.4	Swagger2.0 Definition	85
200	A.6.5	Property Definition	86
201	A.6.6	CRUDN behavior.....	87
202	A.7	Current Temperature Mapping.....	87
203	A.7.1	Introduction	87
204	A.7.2	Example URI	87
205	A.7.3	Resource Type	87
206	A.7.4	Swagger2.0 Definition	87
207	A.7.5	Property Definition	89
208	A.7.6	CRUDN behavior.....	89
209	A.8	Cycle Control Mapping	89
210	A.8.1	Introduction	89
211	A.8.2	Example URI	89
212	A.8.3	Resource Type	90
213	A.8.4	Swagger2.0 Definition	90
214	A.8.5	Property Definition	91
215	A.8.6	CRUDN behavior.....	92
216	A.9	Fan Speed Level Mapping	92
217	A.9.1	Introduction	92
218	A.9.2	Example URI	92
219	A.9.3	Resource Type	92
220	A.9.4	Swagger2.0 Definition	92
221	A.9.5	Property Definition	95
222	A.9.6	CRUDN behavior.....	95
223	A.10	Heating Zone Mapping.....	95
224	A.10.1	Introduction	95
225	A.10.2	Example URI	96
226	A.10.3	Resource Type	96
227	A.10.4	Swagger2.0 Definition	96
228	A.10.5	Property Definition	97
229	A.10.6	CRUDN behavior.....	98
230	A.11	HVAC Fan Mode Mapping	98
231	A.11.1	Introduction	98
232	A.11.2	Example URI	98
233	A.11.3	Resource Type	98
234	A.11.4	Swagger2.0 Definition	98
235	A.11.5	Property Definition	101
236	A.11.6	CRUDN behavior.....	101
237	A.12	On Off Mapping	101
238	A.12.1	Introduction	101

239	A.12.2	Example URI	101
240	A.12.3	Resource Type	101
241	A.12.4	Swagger2.0 Definition	101
242	A.12.5	Property Definition	103
243	A.12.6	CRUDN behavior	103
244	A.13	Oven Cycle Phase Mapping.....	104
245	A.13.1	Introduction	104
246	A.13.2	Example URI	104
247	A.13.3	Resource Type	104
248	A.13.4	Swagger2.0 Definition	104
249	A.13.5	Property Definition	106
250	A.13.6	CRUDN behavior	106
251	A.14	Target Humidity Mapping.....	106
252	A.14.1	Introduction	106
253	A.14.2	Example URI	106
254	A.14.3	Resource Type	106
255	A.14.4	Swagger2.0 Definition	107
256	A.14.5	Property Definition	110
257	A.14.6	CRUDN behavior	111
258	A.15	Target Temperature Mapping	111
259	A.15.1	Introduction	111
260	A.15.2	Example URI	111
261	A.15.3	Resource Type	111
262	A.15.4	Swagger2.0 Definition	111
263	A.15.5	Property Definition	115
264	A.15.6	CRUDN behavior	115
265			
266			

267
268

No table of figures entries found.

Figures

Tables

269		
270	Table 6-1 AllJoyn to OCF Device Type Mapping.....	14
271	Table 7-1 AllJoyn Interface to OCF Resource Type Mapping – Minimum Interface Set	16
272	Table 7-2 AllJoyn Interface to OCF Resource Type Mapping – Optional Interface Set	17
273	Table 8-1 Interface to Resource Summary.....	18
274		
275		

276 **1 Scope**

277 The OCF Resource to AllJoyn Interface Mapping specification (“this specification”) provides
278 detailed mapping information to provide equivalency between AllJoyn defined Interfaces and OCF
279 defined Resources,

280 This specification provides mapping for Device Types (AllJoyn to/from OCF), identifies equivalent
281 OCF Resources for both mandatory and optional AllJoyn interfaces and for each interface defines
282 the detailed Property by Property mapping using OCF defined extensions to JSON schema to
283 programmatically define the mappings.

284

285 **2 Normative references**

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

289 OCF Core Specification, *Open Interconnect Consortium Core Specification*, Version 1.0.

290 OCF Resource Type Specification, *Open Interconnect Consortium Resource Type Specification*,
291 Version 1.0

292 OCF Smart Home Device Specification, *Open Interconnect Consortium Smart Home Device*
293 *Specification*, Version 1.0

294 Derived Models for Interoperability between IoT Ecosystems, Stevens & Merriam, March 2016

295 [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)
296 [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)

297 IETF RFC 7159, *The JavaScript Object Notation (JSON) Data Interchange Format*, March 2014
298 <http://www.ietf.org/rfc/rfc7159.txt>

299 RAML, *Restful API modelling language*, Version 0.8.
300 <https://github.com/raml-org/raml-spec/blob/master/versions/raml-08/raml-08.md>

301 AllJoyn Common Data Model Interface Definitions
302 <https://wiki.alljoyn.org/cdm>

303 Swagger2.0, *Swagger RESTful API Documentation Specification*, Version 2.0
304 <http://swagger.io/specification/>

305 OCF Resource Type Definitions, *API Definition Language for OCF Resource Type Definitions*,
306 Release OCF-v1.0.0
307 <https://github.com/openconnectivityfoundation/derivedmodels>

308

309 **3 Terms, definitions symbols and abbreviations**

310 **3.1 Terms and definitions**

311 **3.1.1**

312 **OCF**

313 Open Connectivity Foundation

314 The organization that created these specifications

315 **3.1.2**
316 **RAML**
317 RESTful API Modelling Language
318 RAML is a simple and succinct way of describing practically-RESTful APIs. See RAML.

319 **3.2 Symbols and abbreviations**
320 None defined.

321 **4 Document conventions and organization**

322 For the purposes of this document, the terms and definitions given in OCF Core Specification and
323 OCF Resource Type Specification apply.

324 **4.1 Conventions**

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

329 **4.2 Notation**

330 In this document, features are described as required, recommended, allowed or DEPRECATED as
331 follows:

332 Required (or shall or mandatory).

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

336 Recommended (or should).

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

344 Allowed (or allowed).

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

348 Conditionally allowed (CA)

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

351 Conditionally required (CR)

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

355 DEPRECATED

356 Although these features are still described in this specification, they should not be implemented
357 except for backward compatibility. The occurrence of a deprecated feature during operation of
358 an implementation compliant with the current specification has no effect on the
359 implementation's operation and does not produce any error conditions. Backward compatibility
360 may require that a feature is implemented and functions as specified but it shall never be used
361 by implementations compliant with this specification.

362 Strings that are to be taken literally are enclosed in "double quotes".

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

364 **4.3 Data types**

365 See OCF Core Specification.

366 **5 Theory of Operation**

367 **5.1 Interworking Approach**

368 The interworking between AllJoyn defined interfaces and OCF defined Resource Types is modelled
369 using the derived model syntax described in Derived Models for Interoperability . Determination
370 of the minimum set of AllJoyn interfaces for which equivalency is required within the OCF data
371 model was done by listing the set of interfaces required for each of the device types defined by
372 the CDM Project inside of AllJoyn. Where the AllJoyn interface supports methods then an
373 actuation design pattern is applied.

374 **5.2 Mapping Syntax**

375 **5.2.1 Block Identification**

376 Within the defined syntax for derived modelling used by this Specification there are two blocks that
377 define the actual Property-Property equivalence or mapping. These blocks are identified by the
378 keywords 'x-to-ocf' and 'x-from-ocf'. Derived Models for Interoperability does not define a rigid
379 syntax for these blocks; they are free form string arrays that contain pseudo-coded mapping logic.
380 Within this specification we apply the rules in the following clauses to these blocks to ensure
381 consistency and re-usability and extensibility of the mapping logic that is defined.

382 **5.2.2 Use of Carriage Return**

383 All statements are terminated with a carriage return.

384 **5.2.3 Value Assignment**

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

387 **5.2.4 Property Naming**

388 All Property names are identical to the name used by the original model; for example from the OCF
389 Temperature Resource the Property name 'temperature' is used whereas when referred to the
390 derived ecosystem then the semantically equivalent Property name is used.

391 When the same name is used by both OCF and the derived ecosystem for semantically equivalent
392 values then the name of the OCF defined Property is prepended by the ecosystem designator 'ocf'
393 to avoid ambiguity (e.g. 'ocf.step')

394 **5.2.5 Arrays**

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

398 **5.2.6 Default Mapping**

399 There are cases where the specified mapping is not possible as one or more of the Properties
400 being mapped is optional in the source model. In all such instances a default mapping is provided.

401 The default map is indicated by the prepending of an 'otherwise:' modifier to the assignment. (e.g.
402 'otherwise: step = 1')

403 **5.2.7 Conditional Mapping**

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

405 if 'condition', 'mapping'.

406 Is applied.

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

408 **5.2.8 Loops**

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

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

412 Where:

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

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

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

416 Is applied.

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

419 **5.2.9 Method Invocation**

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

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

427 **6 Device Type Mapping**

428 **6.1 Introduction**

429 This Section contains the mappings to/from Device Types.

430 **6.2 AllJoyn Device Types to OCF Device Types**

431 The following table captures the equivalency mapping between AllJoyn defined Device Types (see
432 AllJoyn Common Data Model Interface Definitions) and OCF defined Device Types (see Table 10-
433 1 in OCF Smart Home Device Specification). The minimum interface set for the AllJoyn definitions
434 is provided in the HAE Theory of Operation; the minimum Resource sets for each OCF Device is
435 provided in OCF Smart Home Device Specification.

Table 6-1 AllJoyn to OCF Device Type Mapping.

Classification	AllJoyn Device Type	AllJoyn ID	OCF Device Type
Air Care	Air Conditioner	5	oic.d.airconditioner
	Air Purifier	9	oic.d.airpurifier
	Air Quality Monitor	11	oic.d.aqm
	Dehumidifier	8	oic.d.dehumidifier
	Humidifier	7	oic.d.humidifier
	Electric Fan	10	oic.d.fan
	Thermostat	6	oic.d.thermostat
Fabric Care	Clothes Washer	12	oic.d.washer
	Clothes Dryer	13	oic.d.dryer
	Clothes Washer-Dryer	14	oic.d.washerdryer
Food Preservation	Refrigerator	2	oic.d.refrigerator
	Ice-Maker	4	oic.r.icemaker (maps to Resource)
	Freezer	3	oic.d.freezer
Food Preparation	Oven	17	oic.d.oven
	Cooktop	18	oic.d.cooktop
	Cookerhood	19	oic.d.cookerhood
	Food probe	20	oic.d.foodprobe
Dish Care	Dishwasher	15	oic.d.dishwasher
Floor Care	Robot Cleaner	16	oic.d.robotcleaner
Entertainment	Television	21	oic.d.tv
	Set Top Box (STB)	22	oic.d.stb

437

438 **6.3 OCF Device Types with no AllJoyn Equivalent**

439 The following table captures the Device Types defined by OCF have no direct equivalent in AllJoyn,
440 they shall all be mapped to an AllJoyn Device Type of 'Other' (Id of '1').

Table 6-2 OCF Device Types with no AllJoyn Equivalent OCF Device Name	OCF Device Type
Receiver	oic.d.receiver
Blind	oic.d.blind
Door	oic.d.door
Garage Door	oic.d.garagedoor
Generic Sensor	oic.d.sensor
Light	oic.d.light
Smart Plug	oic.d.smartplug
Switch	oic.d.switch
Water Valve	oic.d.watervalve
Printer	oic.d.printer
Multi-Function Printer	oic.d.multifunctionprinter
Scanner	oic.r.scanner
Camera	oic.d.camera
Security Panel	oic.d.securitypanel
Smart Lock	oic.d.smartlock

441

442 **7 Resource to Interface Equivalence**

443 **7.1 Introduction**

444 This Section lists the complete set of applicable AllJoyn Interfaces and provides the equivalent
 445 OCF Resource Type(s) to which the Interfaces map.

446 **7.2 AllJoyn Interfaces to OCF Resources**

447 The following tables capture the equivalency mapping between AllJoyn defined Interfaces (see
 448 AllJoyn Common Data Model Interface Definitions) and OCF defined Resource Types (see OCF
 449 Resource Type Specification). Detailed Property by Property mappings are provided in Section 8.

450 Table 7-1 AllJoyn Interface to OCF Resource Type Mapping – Minimum Interface Set captures the
 451 mappings for Interfaces that are part of the minimum set for an AllJoyn Device.

452 Table 7-2 AllJoyn Interface to OCF Resource Type Mapping – Optional Interface Set captures the
 453 mappings for Interfaces that are optional for an AllJoyn Device; deep translation for these
 454 interfaces via derived modelling is not within the scope of this release of the specification.

455 **Table 7-1 AllJoyn Interface to OCF Resource Type Mapping – Minimum Interface Set**

AllJoyn Interface	OCF Resource Type Name	OCF Resource Type ID	OCF Interface(s)
Environment.CurrentAirQuality	Air Quality Collection	oic.r.airqualitycollection	oic.if.s
Environment.CurrentAirQualityLevel	Air Quality Collection	oic.r.airqualitycollection	oic.if.s
Environment.CurrentHumidity	Humidity	oic.r.humidity	oic.if.s
Environment.CurrentTemperature	Temperature	oic.r.temperature	oic.if.s
Environment.TargetHumidity	Humidity	oic.r.humidity, oic.r.selectablelevels	oic.if.a
Environment.TargetTemperature	Temperature	oic.r.temperature	oic.if.a
Operation.AudioVolume	Audio Controls	oic.r.audio	oic.if.a
Operation.Channel	Not mapped		
Operation.ClimateControlMode	Mode	oic.r.mode	oic.if.a
	Operational State	oic.r.operational.state	oic.if.s
Operation.ClosedStatus	Door	oic.r.door	oic.if.s

Operation.CycleControl	Operational State	oic.r.operational.state	oic.if.s
Operation.FanSpeedLevel	Air Flow	oic.r.airflow	oic.if.a
Operation.HeatingZone	Heating Zone Collection	oic.r.heatingzonecollection	oic.if.s
Operation.HvacFanMode	Mode	oic.r.mode	oic.if.a
Operation.OnOffStatus	Binary Switch	oic.r.switch.binary	oic.if.s
Operation.OvenCyclePhase	Operational State	oic.r.operationalstate	oic.if.s

456

457

Table 7-2 AllJoyn Interface to OCF Resource Type Mapping – Optional Interface Set

AllJoyn Interface	OCF Resource Type Name	OCF Resource Type ID	OCF Interface(s)
Environment.TargetTemperatureLevel	Mode	oic.r.mode	oic.if.a
Environment.WaterLevel	New Resource	TBD	oic.if.s
Environment.WindDirection	Air Flow	oic.r.airflow	oic.if.a
Operation.AirRecirculationMode	Mode	oic.r.mode	oic.if.a
Operation.Alerts	TBD	TBD	TBD
Operation.AudioVideoInput	Media Source List	oic.r.media.input	oic.if.a
Operation.BatteryStatus	Battery	oic.r.energy.battery	oic.if.s
Operation.CurrentPower	Energy Usage	oic.r.energy.usage	oic.if.s
Operation.DishWashingCyclePhase	Operational State	oic.r.operationalstate	oic.if.s
Operation.EnergyUsage	Energy Usage	oic.r.energy.usage	oic.if.s

Operation.FilterStatus	New Resource	TBD	TBD
Operation.LaundryCyclePhase	Mode	oic.r.mode	oic.if.s
Operation.MoistureOutputLevel	Mode	oic.r.mode	oic.if.a
Operation.PlugInUnits	TBD	TBD	TBD
Operation.RapidMode	Refrigeration	oic.r.refrigeration	oic.if.a
Operation.RemoteControllability	TBD	TBD	TBD
Operation.RepeatMode	Ecomode	oic.r.ecomode	oic.if.a
Operation.ResourceSaving	New Resource	TBD	TBD
Operation.RobotCleaningCyclePhase	Mode	oic.r.mode	oic.if.s
Operation.SoilLevel	Mode	oic.r.mode	oic.if.a
Operation.SpinSpeedLevel	Mode	oic.r.mode	oic.if.a
Operation.Timer	Time Period	oic.r.time.period	oic.if.s

458

459

460 8 Detailed Mapping APIs

461 8.1 Introduction

462 This section provides an API (using RAML) and a mapping description (using JSON that aligns
463 with the Derived Modelling syntax described in [Derived Model White Paper]) for all Interfaces and
464 Resources that are within scope.

465 Annex A provides definitions for all mappings making use of Swagger2.0 instead of RAML and
466 JSON.

467 All the sub-clauses in clause 8 and Annex A describe the Resource Types with a restful API
468 definition language. The Resource Type definitions presented in clause 8 and Annex A are
469 formatted for readability, and so may appear to have extra line breaks. The contents of the
470 Resource Types without the extra line breaks are available in OCF Resource Type Definitions.

471 Table 8-1 Interface to Resource Summary provides a reference and link to the per Interface sub-
472 sections.

473

Table 8-1 Interface to Resource Summary

474

AllJoyn Interface Name	Equivalent Resource(s)	Mapping Section
Environment.CurrentAirQuality	oic.r.airqualitycollection	8.2
Environment.CurrentAirQualityLevel	oic.r.airqualitycollection	8.3
Environment.CurrentHumidity	oic.r.humidity	8.4
Environment.CurrentTemperature	oic.r.temperature	8.5
Environment.TargetHumidity	oic.r.humidity, oic.r.selectablelevels	8.6
Environment.TargetTemperature	oic.r.temperature	8.7
Operation.AudioVolume	oic.r.audio	8.8
Operation.ClimateControlMode	oic.r.mode, oic.r.operationalstate	8.9
Operation.ClosedStatus	oic.r.door	8.10
Operation.CycleControl	oic.r.operational.state	8.11
Operation.FanSpeedLevel	oic.r.airflow	8.12
Operation.HeatingZone	oic.r.heatingzonecollection	8.13
Operation.HvacFanMode	oic.r.mode	8.14
Operation.OnOffStatus, Operation.OnControl, Operation.OffControl	oic.r.switch.binary	8.15
Operation.OvenCyclePhase	oic.r.operationalstate	8.16

475

476 **8.2 Air Quality Mapping**

477 **8.2.1 Introduction**

478 This API defines the mapping between the AllJoyn AirQuality interface and the OCF AirQuality
479 Resource. If more than one instance of the AirQuality interface is exposed then each instance
480 maps to an instance of the OCF AirQuality Resource. The mapping defined in the schema
481 describes the population of the OCF AirQuality Resource. Even if there is only a single instance
482 of an OCF AirQuality Resource this shall be included in an instance of an OCF AirQualityCollection.
483 The number of links in the collection equates to the number of instances of the AllJoyn
484 CurrentAirQuality interface that are exposed. When mapping from OCF the valueType of the
485 Resource shall be introspected, this API is invoked only if this is set to 'Measured'

486 **8.2.2 Example URI**

487 /CurrentAirQualityResURI

488 **8.2.3 Resource Type**

489 The resource type (rt) is defined as: oic.r.airqualitycollection.

490 **8.2.4 RAML Definition**

```
491 #%RAML 0.8
492 title: CurrentAirQualityInterfaceMapping
493 version: OCFv1.0.0-20170317
494 traits:
495   - interface-sensor :
496     queryParameters:
497       if:
498         enum: ["oic.if.s", "oic.if.baseline"]
499
500 /CurrentAirQualityResURI:
501   description: |
502     This API defines the mapping between the AllJoyn AirQuality interface and the OCF AirQuality
503     Resource.
504     If more than one instance of the AirQuality interface is exposed then each instance maps to an
505     instance of the OCF AirQuality Resource.
506     The mapping defined in the schema describes the population of the OCF AirQuality Resource.
507     Even if there is only a single instance of an OCF AirQuality Resource this shall be included in
508     an instance of an OCF AirQualityCollection.
509     The number of links in the collection equates to the number of instances of the AllJoyn
510     CurrentAirQuality interface that are exposed.
511     When mapping from OCF the valueType of the Resource shall be introspected, this API is invoked
512     only if this is set to 'Measured'
513
514   is : ['interface-sensor']
515   get:
516     responses :
517       200:
518         body:
519           application/json:
520             schema: /
521               {
522                 "id":
523 "http://openinterconnect.org/asamapping/schemas/asa.environment.currentairquality.json#",
524                 "$schema": "http://json-schema.org/draft-04/schema#",
525                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
526 reserved.",
527                 "title": "Current Air Quality",
528                 "definitions": {
529                   "asa.environment.currentairquality": {
530                     "type": "object",
531                     "properties": {
532                       "contaminanttype": {
533                         "type": "integer",
534                         "description": "The contaminant type",
535                         "x-ocf-conversion": {
536                           "x-ocf-alias": "oic.r.airquality",
537                           "x-to-ocf": [
538                             "valuetype = Measured",
539                             "contaminanttypearray = [CH2O,CO2,CO,PM2_5,PM10,VOC]",
540                             "ocf.contaminanttype = contaminanttypearray[contaminanttype]"
541                           ],
542                           "x-from-ocf": [
543                             "contaminanttype = indexof contaminanttypearray[ocf.contaminanttype]"
544                           ]
545                         }
546                       }
547                     }
548                   }
549                 }
550               }
```

```

546     },
547     "currentvalue": {
548       "type": "number",
549       "x-ocf-conversion": {
550         "x-ocf-alias": "oic.r.airquality",
551         "x-to-ocf": [
552           "contaminantvalue = currentvalue"
553         ],
554         "x-from-ocf": [
555           "currentvalue = contaminantvalue"
556         ]
557       }
558     },
559     "minvalue": {
560       "type": "number",
561       "x-ocf-conversion": {
562         "x-ocf-alias": "oic.r.airquality",
563         "x-to-ocf": [
564           "range[0] = minvalue"
565         ],
566         "x-from-ocf": [
567           "minvalue = range[0]"
568         ]
569       }
570     },
571     "maxvalue": {
572       "type": "number",
573       "x-ocf-conversion": {
574         "x-ocf-alias": "oic.r.airquality",
575         "x-to-ocf": [
576           "range[1] = maxvalue"
577         ],
578         "x-from-ocf": [
579           "maxvalue = range[1]"
580         ]
581       }
582     },
583     "precision": {
584       "type": "number",
585       "x-ocf-conversion": {
586         "x-ocf-alias": "oic.r.airquality",
587         "x-to-ocf": [
588           "ocf.precision = precision"
589         ],
590         "x-from-ocf": [
591           "precision = ocf.precision"
592         ]
593       }
594     },
595     "updatemintime": {
596       "type": "integer",
597       "x-ocf-conversion": {
598         "x-ocf-alias": "oic.r.value.conditional",
599         "x-to-ocf": [
600           "ocf.minnotifyperiod = updatemintime"
601         ],
602         "x-from-ocf": [
603           "updatemintime = ocf.minnotifyperiod"
604         ]
605       }
606     }
607   }
608 }
609 },
610 "type": "object",
611 "allOf": [
612   {"$ref": "#/definitions/asa.environment.currentairquality"}
613 ],
614 "required":
615 ["contaminanttype", "currentvalue", "minvalue", "maxvalue", "precision", "updatemintime"]

```

```

616     }
617
618     example: /
619     {
620       "rt":      ["oic.r.airqualitycollection"]
621     }
622

```

8.2.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
currentvalue	oic.r.airquality	contaminantvalue = currentvalue	currentvalue = contaminantvalue	
updateinterval	oic.r.value.conditional	ocf.minnotifyperiod = updateinterval	updateinterval = ocf.minnotifyperiod	
maxvalue	oic.r.airquality	range[1] = maxvalue	maxvalue = range[1]	
precision	oic.r.airquality	ocf.precision = precision	precision = ocf.precision	
minvalue	oic.r.airquality	range[0] = minvalue	minvalue = range[0]	
contaminanttype	oic.r.airquality	valuetype = Measuredcontaminanttypearray = [CH2O,CO2,CO,PM2_5,PM10,VOC]ocf.contaminanttype = contaminanttypearray[contaminanttype]	contaminanttype = indexof contaminanttypearray[ocf.contaminanttype]	The contaminant type

8.2.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentAirQualityResURI		get			

8.3 Air Quality Level Mapping

8.3.1 Introduction

627 This API defines the mapping between the AllJoyn AirQualityLevel interface and the OCF
628 AirQuality Resource. If more than one instance of the AirQualityLevel interface is exposed then
629 each instance maps to an instance of the OCF AirQuality Resource. The mapping defined in the
630 schema describes the population of the OCF AirQuality Resource. Even if there is only a single
631 instance of an OCF AirQuality Resource then this shall be included in an instance of an OCF
632 AirQualityCollection. The number of links in the collection equates to the number of instances of
633 the AllJoyn CurrentAirQuality interface that are exposed. When mapping from OCF the valueType
634 of the Resource shall be introspected, this API is invoked only if this is set to 'Qualitative'

8.3.2 Example URI

636 /CurrentAirQualityLevelResURI

8.3.3 Resource Type

638 The resource type (rt) is defined as: oic.r.airqualitycollection.

8.3.4 RAML Definition

640 `##RAML 0.8`

641 `title: CurrentAirQualityLevelInterfaceMapping`

642 `version: OCFv1.0.0-20170317`

```

643 traits:
644   - interface-sensor :
645     queryParameters:
646       if:
647         enum: ["oic.if.s", "oic.if.baseline"]
648
649 /CurrentAirQualityLevelResURI:
650   description: |
651     This API defines the mapping between the AllJoyn AirQualityLevel interface and the OCF
652     AirQuality Resource.
653     If more than one instance of the AirQualityLevel interface is exposed then each instance maps
654     to an instance of the OCF AirQuality Resource.
655     The mapping defined in the schema describes the population of the OCF AirQuality Resource.
656     Even if there is only a single instance of an OCF AirQuality Resource then this shall be
657     included in an instance of an OCF AirQualityCollection.
658     The number of links in the collection equates to the number of instances of the AllJoyn
659     CurrentAirQuality interface that are exposed.
660     When mapping from OCF the valueType of the Resource shall be introspected, this API is invoked
661     only if this is set to 'Qualitative'
662
663   is : ['interface-sensor']
664
665   get:
666     responses :
667       200:
668         body:
669           application/json:
670             schema: /
671               {
672                 "id":
673                 "http://openinterconnect.org/asamapping/schemas/asa.environment.currentairqualitylevel.json#",
674                 "$schema": "http://json-schema.org/draft-04/schema#",
675                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
676                 reserved.",
677                 "title": "Current Air Quality Level",
678                 "definitions": {
679                   "asa.environment.currentairqualitylevel": {
680                     "type": "object",
681                     "properties": {
682                       "contaminanttype": {
683                         "type": "integer",
684                         "description": "The contaminant type",
685                         "x-ocf-conversion": {
686                           "x-ocf-alias": "oic.r.airquality",
687                           "x-to-ocf": [
688                             "valuetype = Qualitative",
689                             "if contaminanttype = 0, ocf.contaminanttype = CH20",
690                             "if contaminanttype = 1, ocf.contaminanttype = CO2",
691                             "if contaminanttype = 2, ocf.contaminanttype = CO",
692                             "if contaminanttype = 3, ocf.contaminanttype = PM2_5",
693                             "if contaminanttype = 4, ocf.contaminanttype = PM10",
694                             "if contaminanttype = 5, ocf.contaminanttype = VOC",
695                             "if contaminanttype = 253, ocf.contaminanttype = Smoke",
696                             "if contaminanttype = 254, ocf.contaminanttype = Odor",
697                             "if contaminanttype = 255, ocf.contaminanttype = AirPollution"
698                           ],
699                           "x-from-ocf": [
700                             "if ocf.contaminanttype = CH20, contaminanttype = 0",
701                             "if ocf.contaminanttype = CO2, contaminanttype = 1",
702                             "if ocf.contaminanttype = CO, contaminanttype = 2",
703                             "if ocf.contaminanttype = PM2_5, contaminanttype = 3",
704                             "if ocf.contaminanttype = PM10, contaminanttype = 4",
705                             "if ocf.contaminanttype = VOC, contaminanttype = 5",
706                             "if ocf.contaminanttype = Smoke, contaminanttype = 253",
707                             "if ocf.contaminanttype = Odor, contaminanttype = 254",
708                             "if ocf.contaminanttype = AirPollution, contaminanttype = 255"

```



```

708     ]
709   }
710 },
711   "currentlevel": {
712     "type": "integer",
713     "x-ocf-conversion": {
714       "x-ocf-alias": "oic.r.airquality",
715       "x-to-ocf": [
716         "contaminantvalue = currentlevel"
717       ],
718       "x-from-ocf": [
719         "currentlevel = contaminantvalue"
720       ]
721     }
722   },
723   "maxlevel": {
724     "type": "integer",
725     "x-ocf-conversion": {
726       "x-ocf-alias": "oic.r.airquality",
727       "x-to-ocf": [
728         "range[0] = 0",
729         "range[1] = maxvalue"
730       ],
731       "x-from-ocf": [
732         "maxvalue = range[1]"
733       ]
734     }
735   }
736 }
737 }
738 },
739 "type": "object",
740 "allof": [
741   {"$ref": "#/definitions/asa.environment.currentairqualitylevel"}
742 ],
743 "required": ["contaminanttype", "currentlevel", "maxlevel"]
744 }
745
746 example: /
747 {
748   "rt": ["oic.r.airqualitycollection"]
749 }
750

```

8.3.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
currentlevel	oic.r.airquality	contaminantvalue = currentlevel	currentlevel = contaminantvalue	
maxlevel	oic.r.airquality	range[0] = 0range[1] = maxvalue	maxvalue = range[1]	
contaminanttype	oic.r.airquality	valuetype = Qualitativeif contaminanttype = 0, ocf.contaminanttype = CH2Oif contaminanttype = 1, ocf.contaminanttype = CO2if contaminanttype = 2, ocf.contaminanttype	if ocf.contaminanttype = CH2O, contaminanttype = 0if ocf.contaminanttype = CO2, contaminanttype = 1if ocf.contaminanttype = CO, contaminanttype = 2if	The contaminant type

		= COif contaminanttype = 3, ocf.contaminanttype = PM2_5if contaminanttype = 4, ocf.contaminanttype = PM10if contaminanttype = 5, ocf.contaminanttype = VOCif contaminanttype = 253, ocf.contaminanttype = Smokeif contaminanttype = 254, ocf.contaminanttype = Odorif contaminanttype = 255, ocf.contaminanttype = AirPollution	ocf.contaminanttype = PM2_5, contaminanttype = 3if ocf.contaminanttype = PM10, contaminanttype = 4if ocf.contaminanttype = VOC, contaminanttype = 5if ocf.contaminanttype = Smoke, contaminanttype = 253if ocf.contaminanttype = Odor, contaminanttype = 254if ocf.contaminanttype = AirPollution, contaminanttype = 255	
--	--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

752 **8.3.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/CurrentAirQualityLevelResURI		get			

753 **8.4 Current Humidity Mapping**

754 **8.4.1 Introduction**

755 This API defines the mapping between an instance of an OCF Humidity which exposes only a
 756 sensor interface and the AllJoyn Current Humidity interface. A RETRIEVE on a Temperature
 757 Sensor maps to an action on an instance of an Environment.CurrentTemperature Interface.

758 **8.4.2 Example URI**

759 /CurrentHumidityResURI

760 **8.4.3 Resource Type**

761 The resource type (rt) is defined as: oic.r.humidity.

762 **8.4.4 RAML Definition**

```

763 #%RAML 0.8
764 title: CurrentHumidityInterfaceMapping
765 version: OCFv1.0.0-20170317
766 traits:
767   - interface-sensor :
768     queryParameters:
769       if:
770         enum: ["oic.if.s", "oic.if.baseline"]
771
772 /CurrentHumidityResURI:
773   description: |
774     This API defines the mapping between an instance of an OCF Humidity which exposes only a sensor
775     interface
  
```

```

776     and the AllJoyn Current Humidity interface.
777     A RETRIEVE on a Temperature Sensor maps to an action on an instance of an
778     Environment.CurrentTemperature Interface.
779
780     is : ['interface-sensor']
781
782     get:
783         responses :
784             200:
785                 body:
786                     application/json:
787                         schema: /
788                             {
789                                 "id":
790                                 "http://openinterconnect.org/asamapping/schemas/asa.environment.currenthumidity.json#",
791                                 "$schema": "http://json-schema.org/draft-04/schema#",
792                                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
793                                 reserved.",
794                                 "title": "Current Humidity",
795                                 "definitions": {
796                                     "asa.environment.currenthumidity": {
797                                         "type": "object",
798                                         "properties": {
799                                             "currentvalue": {
800                                                 "type": "number",
801                                                 "description": "Measured value",
802                                                 "x-ocf-conversion": {
803                                                     "x-ocf-alias": "oic.r.humidity",
804                                                     "x-to-ocf": [
805                                                         "humidity = currentValue"
806                                                     ],
807                                                     "x-from-ocf": [
808                                                         "currentvalue = humidity"
809                                                     ]
810                                                 },
811                                             "maxvalue": {
812                                                 "type": "number",
813                                                 "description": "Max measured value for humidity",
814                                                 "x-ocf-conversion": {
815                                                     "x-ocf-alias": "oic.r.humidity",
816                                                     "x-to-ocf": [
817                                                         "range[0] = 0",
818                                                         "range[1] = maxvalue"
819                                                     ],
820                                                     "x-from-ocf": [
821                                                         "maxvalue = range[1]"
822                                                     ]
823                                                 },
824                                         }
825                                     }
826                                 },
827                                 "type": "object",
828                                 "allOf": [
829                                     {"$ref": "#/definitions/asa.environment.currenthumidity"}
830                                 ],
831                                 "required": [ "currentvalue", "maxvalue" ]
832                             }
833
834
835         example: /
836             {
837                 "rt":      ["oic.r.humidity"]
838             }
839

```

840 **8.4.5 Property Definition**

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
currentvalue	oic.r.humidity	humidity currentValue	= currentvalue humidity	= Measured value
maxvalue	oic.r.humidity	range[0] Orange[1] maxvalue	= = maxvalue range[1]	= Max measured value for humidty

841 **8.4.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/CurrentHumidityResURI		get			

842 **8.5 Current Temperature Mapping**

843 **8.5.1 Introduction**

844 This API defines the mapping between an instance of an OCF Temperature which exposes only a
 845 sensor interface and the AllJoyn Current Temperature interface. A RETRIEVE on a Temperature
 846 Sensor maps to an action on an instance of an Environment.CurrentTemperature Interface.

847 **8.5.2 Example URI**

848 /CurrentTemperatureResURI

849 **8.5.3 Resource Type**

850 The resource type (rt) is defined as: oic.r.temperature.

851 **8.5.4 RAML Definition**

```

852 #%RAML 0.8
853 title: CurrentTemperatureInterfaceMapping
854 version: OCFv1.0.0-20170317
855 traits:
856   - interface-sensor :
857     queryParameters:
858       if:
859         enum: ["oic.if.s", "oic.if.baseline"]
860
861 /CurrentTemperatureResURI:
862   description: |
863     This API defines the mapping between an instance of an OCF Temperature which exposes only a
864     sensor interface
865     and the AllJoyn Current Temperature interface.
866     A RETRIEVE on a Temperature Sensor maps to an action on an instance of an
867     Environment.CurrentTemperature Interface.
868
869   is : ['interface-sensor']
870   get:
871     responses :
872       200:
873         body:
874           application/json:
875             schema: /
876               {
877                 "id":
878                 "http://openinterconnect.org/asamapping/schemas/asa.environment.currenttemperature.json#",
879                 "$schema": "http://json-schema.org/draft-04/schema#",
880                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
881                 reserved.",

```

```

882 "title": "Current Temperature",
883 "definitions": {
884   "asa.environment.currenttemperature": {
885     "type": "object",
886     "properties": {
887       "currentvalue": {
888         "type": "number",
889         "description": "Measured value",
890         "x-ocf-conversion": {
891           "x-ocf-alias": "oic.r.temperature",
892           "x-to-ocf": [
893             "temperature = currentValue",
894             "units = C"
895           ],
896           "x-from-ocf": {
897             "oneOf": [
898               {
899                 "properties": {
900                   "units": "string",
901                   "enum": ["C"]
902                 },
903                 "x-from-ocf": [
904                   "currentvalue = temperature"
905                 ]
906               },
907               {
908                 "properties": {
909                   "units": "string",
910                   "enum": ["F"]
911                 },
912                 "x-from-ocf": [
913                   "currentvalue = (temperature-32)*5/9"
914                 ]
915               },
916               {
917                 "properties": {
918                   "units": "string",
919                   "enum": ["K"]
920                 },
921                 "x-from-ocf": [
922                   "currentvalue = temperature-273.15"
923                 ]
924               }
925             ]
926           }
927         },
928         "precision": {
929           "type": "number",
930           "x-ocf-conversion": {
931             "x-ocf-alias": "oic.r.temperature",
932             "x-to-ocf": [
933               "ocf.precision = precision"
934             ],
935             "x-from-ocf": [
936               "precision = ocf.precision"
937             ]
938           }
939         },
940         "updatemintime": {
941           "type": "integer",
942           "x-ocf-conversion": {
943             "x-ocf-alias": "oic.r.value.conditional",
944             "x-to-ocf": [
945               "ocf.minnotifyperiod = updatemintime"
946             ],
947             "x-from-ocf": [
948               "updatemintime = ocf.minnotifyperiod"
949             ]
950           }
951         }
952       }

```

```

953     }
954   },
955 },
956 "type": "object",
957 "allOf": [
958   { "$ref": "#/definitions/asa.environment.currenttemperature" }
959 ],
960 "required": [ "currentvalue", "precision", "updatemintime" ]
961 }
962
963 example: /
964 {
965   "rt":      ["oic.r.temperature"]
966 }
967

```

8.5.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
currentvalue	oic.r.temperature	temperature = currentValueunits = C	oneOf	Measured value
updatemintime	oic.r.value.conditiona l	ocf.minnotifyperio d = updatemintime	updatemintime = ocf.minnotifyperio d	
precision	oic.r.temperature	ocf.precision = precision	precision = ocf.precision	

8.5.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentTemperatureResURI		get			

8.6 Target Humidity Mapping

8.6.1 Introduction

This API defines the mapping between an instance of an AllJoyn TargetHumidity Interface and the OCF Resource Equivalent. A POST on a Humidity Sensor maps to an action on an instance of an Environment.TargetHumidity Interface.

8.6.2 Example URI

/TargetHumidityResURI

8.6.3 Resource Type

The resource type (rt) is defined as: oic.r.humidity.

8.6.4 RAML Definition

```

980 #%RAML 0.8
981 title: TargetHumidityInterfaceMapping
982 version: OCFv1.0.0-20170317
983 traits:
984   - interface-actuator :
985     queryParameters:
986       if:
987         enum: ["oic.if.a", "oic.if.baseline"]
988
989 /TargetHumidityResURI:
990   description: |

```

```

991     This API defines the mapping between an instance of an AllJoyn TargetHumidity Interface and the
992 OCF Resource Equivalent.
993     A POST on a Humidity Sensor maps to an action on an instance of an Environment.TargetHumidity
994 Interface.
995
996     is : ['interface-actuator']
997
998     get:
999         responses :
1000             200:
1001                 body:
1002                     application/json:
1003                         schema: /
1004                             {
1005                                 "id":
1006 "http://openinterconnect.org/asamapping/schemas/asa.environment.targethumidity.json#",
1007                                 "$schema": "http://json-schema.org/draft-04/schema#",
1008                                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1009 reserved.",
1010                                 "title": "Target Humidity",
1011                                 "definitions": {
1012                                     "asa.environment.targethumidity": {
1013                                         "type": "object",
1014                                         "properties": {
1015                                             "targetvalue": {
1016                                                 "type": "number",
1017                                                 "description": "Measured value",
1018                                                 "x-ocf-conversion": {
1019                                                     "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
1020                                                     "x-to-ocf": [
1021 "if minvalue != maxvalue, ocf.desiredhumidity =
1022 targetvalue;ocf.targetlevel = selectablehumiditylevels[0].",
1023 "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
1024 ],
1025 "x-from-ocf": [
1026 "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
1027 "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
1028 ]
1029 }
1030 },
1031 "minvalue": {
1032     "type": "number",
1033     "x-ocf-conversion": {
1034         "x-ocf-alias": "oic.r.humidity",
1035         "x-to-ocf": [
1036             "range[0] = minvalue"
1037         ],
1038         "x-from-ocf": [
1039             "minvalue = range[0]",
1040             "otherwise: minvalue = 0"
1041         ]
1042     }
1043 },
1044 "maxvalue": {
1045     "type": "number",
1046     "x-ocf-conversion": {
1047         "x-ocf-alias": "oic.r.humidity",
1048         "x-to-ocf": [
1049             "range[1] = maxvalue"
1050         ],
1051         "x-from-ocf": [
1052             "maxvalue = range[1]",
1053             "otherwise: maxvalue = 100"
1054         ]
1055     }
1056 },
1057 "stepvalue": {
1058     "type": "number",
1059     "x-ocf-conversion": {

```

```

1059         "x-ocf-alias": "oic.r.humidity",
1060         "x-to-ocf": [
1061             "step = stepvalue"
1062         ],
1063         "x-from-ocf": [
1064             "stepvalue = step",
1065             "otherwise: stepvalue = 1"
1066         ]
1067     },
1068 },
1069     "selectablehumiditylevels": {
1070         "type": "array",
1071         "items": {
1072             "type": "number"
1073         },
1074         "x-ocf-conversion": {
1075             "x-ocf-alias": "oic.r.selectablelevels",
1076             "x-to-ocf": [
1077                 "availablelevels[] = selectablehumiditylevels[]"
1078             ],
1079             "x-from-ocf": [
1080                 "selectablehumiditylevels[] = availablelevels[]"
1081             ]
1082         }
1083     }
1084 },
1085 },
1086 },
1087     "type": "object",
1088     "allOf": [
1089         {"$ref": "#/definitions/asa.environment.targethumidity"}
1090     ],
1091     "required":
1092 [ "targetvalue", "minvalue", "maxvalue", "stepvalue", "selectablehumiditylevels" ]
1093     }
1094
1095     example: /
1096     {
1097         "rt":      ["oic.r.humidity", "oic.r.selectablelevels"]
1098     }
1099
1100     post:
1101     body:
1102     application/json:
1103     schema: /
1104     {
1105         "id":
1106 "http://openinterconnect.org/asamapping/schemas/asa.environment.targethumidity.json#",
1107         "$schema": "http://json-schema.org/draft-04/schema#",
1108         "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1109 reserved.",
1110         "title": "Target Humidity",
1111         "definitions": {
1112             "asa.environment.targethumidity": {
1113                 "type": "object",
1114                 "properties": {
1115                     "targetvalue": {
1116                         "type": "number",
1117                         "description": "Measured value",
1118                         "x-ocf-conversion": {
1119                             "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
1120                             "x-to-ocf": [
1121                                 "if minvalue != maxvalue, ocf.desiredhumidity = targetvalue;ocf.targetlevel
1122 = selectablehumiditylevels[0].",
1123                                 "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
1124                             ],
1125                             "x-from-ocf": [
1126                                 "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",

```



```

1127         "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
1128     ]
1129     }
1130 },
1131 "minvalue": {
1132     "type": "number",
1133     "x-ocf-conversion": {
1134         "x-ocf-alias": "oic.r.humidity",
1135         "x-to-ocf": [
1136             "range[0] = minvalue"
1137         ],
1138         "x-from-ocf": [
1139             "minvalue = range[0]",
1140             "otherwise: minvalue = 0"
1141         ]
1142     }
1143 },
1144 "maxvalue": {
1145     "type": "number",
1146     "x-ocf-conversion": {
1147         "x-ocf-alias": "oic.r.humidity",
1148         "x-to-ocf": [
1149             "range[1] = maxvalue"
1150         ],
1151         "x-from-ocf": [
1152             "maxvalue = range[1]",
1153             "otherwise: maxvalue = 100"
1154         ]
1155     }
1156 },
1157 "stepvalue": {
1158     "type": "number",
1159     "x-ocf-conversion": {
1160         "x-ocf-alias": "oic.r.humidity",
1161         "x-to-ocf": [
1162             "step = stepvalue"
1163         ],
1164         "x-from-ocf": [
1165             "stepvalue = step",
1166             "otherwise: stepvalue = 1"
1167         ]
1168     }
1169 },
1170 "selectablehumiditylevels": {
1171     "type": "array",
1172     "items": {
1173         "type": "number"
1174     },
1175     "x-ocf-conversion": {
1176         "x-ocf-alias": "oic.r.selectablelevels",
1177         "x-to-ocf": [
1178             "availablelevels[] = selectablehumiditylevels[]"
1179         ],
1180         "x-from-ocf": [
1181             "selectablehumiditylevels[] = availablelevels[]"
1182         ]
1183     }
1184 }
1185 }
1186 }
1187 },
1188 "type": "object",
1189 "allOf": [
1190     {"$ref": "#/definitions/asa.environment.targethumidity"}
1191 ],
1192 "required":
1193 [ "targetvalue", "minvalue", "maxvalue", "stepvalue", "selectablehumiditylevels" ]
1194 }
1195
1196     responses :

```

```

1197     200:
1198         body:
1199             application/json:
1200                 schema: /
1201                     {
1202                         "id":
1203 "http://openinterconnect.org/asamapping/schemas/asa.environment.targethumidity.json#",
1204                         "$schema": "http://json-schema.org/draft-04/schema#",
1205                         "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1206 reserved.",
1207                         "title": "Target Humidity",
1208                         "definitions": {
1209                             "asa.environment.targethumidity": {
1210                                 "type": "object",
1211                                 "properties": {
1212                                     "targetvalue": {
1213                                         "type": "number",
1214                                         "description": "Measured value",
1215                                         "x-ocf-conversion": {
1216                                             "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
1217                                             "x-to-ocf": [
1218 "if minvalue != maxvalue, ocf.desiredhumidity =
1219 targetvalue;ocf.targetlevel = selectablehumiditylevels[0].",
1220 "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
1221 ],
1222 "x-from-ocf": [
1223 "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
1224 "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
1225 ]
1226 }
1227 },
1228 "minvalue": {
1229     "type": "number",
1230     "x-ocf-conversion": {
1231         "x-ocf-alias": "oic.r.humidity",
1232         "x-to-ocf": [
1233             "range[0] = minvalue"
1234         ],
1235         "x-from-ocf": [
1236             "minvalue = range[0]",
1237             "otherwise: minvalue = 0"
1238         ]
1239     }
1240 },
1241 "maxvalue": {
1242     "type": "number",
1243     "x-ocf-conversion": {
1244         "x-ocf-alias": "oic.r.humidity",
1245         "x-to-ocf": [
1246             "range[1] = maxvalue"
1247         ],
1248         "x-from-ocf": [
1249             "maxvalue = range[1]",
1250             "otherwise: maxvalue = 100"
1251         ]
1252     }
1253 },
1254 "stepvalue": {
1255     "type": "number",
1256     "x-ocf-conversion": {
1257         "x-ocf-alias": "oic.r.humidity",
1258         "x-to-ocf": [
1259             "step = stepvalue"
1260         ],
1261         "x-from-ocf": [
1262             "stepvalue = step",
1263             "otherwise: stepvalue = 1"
1264         ]
1265     }
1266 }

```

```

1266     },
1267     "selectablehumiditylevels": {
1268       "type": "array",
1269       "items": {
1270         "type": "number"
1271       },
1272       "x-ocf-conversion": {
1273         "x-ocf-alias": "oic.r.selectablelevels",
1274         "x-to-ocf": [
1275           "availablelevels[] = selectablehumiditylevels[]"
1276         ],
1277         "x-from-ocf": [
1278           "selectablehumiditylevels[] = availablelevels[]"
1279         ]
1280       }
1281     }
1282   }
1283 },
1284 "type": "object",
1285 "allOf": [
1286   {"$ref": "#/definitions/asa.environment.targethumidity"}
1287 ],
1288 "required":
1289 [ "targetvalue", "minvalue", "maxvalue", "stepvalue", "selectablehumiditylevels" ]
1290 }
1291 }
1292
1293 example: /
1294 {
1295   "rt":      ["oic.r.humidity", "oic.r.selectablelevels"]
1296 }
1297

```

1298 8.6.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
stepvalue	oic.r.humidity	step = stepvalue	stepvalue = stepotherwise: stepvalue = 1	
targetvalue	oic.r.humidity,oic.r.selectablelevels	if minvalue != maxvalue, ocf.desiredhumidity = targetvalue;ocf.targetlevel = selectablehumiditylevels[0].if minvalue == maxvalue, ocf.targetlevel = targetvalue.	if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel.	Measured value
maxvalue	oic.r.humidity	range[1] = maxvalue	maxvalue = range[1]otherwise: maxvalue = 100	
selectablehumiditylevels	oic.r.selectablelevels	availablelevels[] = selectablehumiditylevels[]	selectablehumiditylevels[] = availablelevels[]	
minvalue	oic.r.humidity	range[0] = minvalue	minvalue = range[0]otherwise: minvalue = 0	

1299 **8.6.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/TargetHumidityResURI		get	post		

1300 **8.7 Target Temperature Mapping**

1301 **8.7.1 Introduction**

1302 This API defines the mapping between an instance of an OCF Temperature which exposes only a
 1303 sensor interface and the AllJoyn Current Temperature interface. A RETRIEVE on a Temperature
 1304 Sensor maps to an action on an instance of an Environment.CurrentTemperature Interface.

1305 **8.7.2 Example URI**

1306 /TargetTemperatureResURI

1307 **8.7.3 Resource Type**

1308 The resource type (rt) is defined as: oic.r.temperature.

1309 **8.7.4 RAML Definition**

```

1310 #%RAML 0.8
1311 title: TargetTemperatureInterfaceMapping
1312 version: OCFv1.0.0-20170317
1313 traits:
1314   - interface-actuator :
1315       queryParameters:
1316
1317         if:
1318           enum: ["oic.if.a", "oic.if.baseline"]
1319
1320 /TargetTemperatureResURI:
1321   description: |
1322     This API defines the mapping between an instance of an OCF Temperature which exposes only a
1323     sensor interface
1324     and the AllJoyn Current Temperature interface.
1325     A RETRIEVE on a Temperature Sensor maps to an action on an instance of an
1326     Environment.CurrentTemperature Interface.
1327
1328   is : ['interface-actuator']
1329   get:
1330     responses :
1331       200:
1332         body:
1333           application/json:
1334             schema: /
1335             {
1336               "id":
1337               "http://openinterconnect.org/asamapping/schemas/asa.environment.targettemperature.json#",
1338               "$schema": "http://json-schema.org/draft-04/schema#",
1339               "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1340               reserved.",
1341               "title": "Target Temperature",
1342               "definitions": {
1343                 "asa.environment.targettemperature": {
1344                   "type": "object",
1345                   "properties": {
1346                     "targetvalue": {
1347                       "type": "number",
1348                       "description": "Measured value",
1349                       "x-ocf-conversion": {
1350                         "x-ocf-alias": "oic.r.temperature",
1351                         "x-to-ocf": [
1352                           "temperature = targetvalue",

```

```

1352         "units = C"
1353     ],
1354     "x-from-ocf": {
1355         "oneOf": [
1356             {
1357                 "properties": {
1358                     "units": "string",
1359                     "enum": ["C"]
1360                 },
1361                 "x-from-ocf": [
1362                     "targetvalue = temperature"
1363                 ]
1364             },
1365             {
1366                 "properties": {
1367                     "units": "string",
1368                     "enum": ["F"]
1369                 },
1370                 "x-from-ocf": [
1371                     "targetvalue = (temperature-32)*5/9"
1372                 ]
1373             },
1374             {
1375                 "properties": {
1376                     "units": "string",
1377                     "enum": ["K"]
1378                 },
1379                 "x-from-ocf": [
1380                     "targetvalue = temperature-273.15"
1381                 ]
1382             }
1383         ]
1384     }
1385 },
1386 "minvalue": {
1387     "type": "number",
1388     "x-ocf-conversion": {
1389         "x-ocf-alias": "oic.r.temperature",
1390         "x-to-ocf": [
1391             "range[0] = minvalue"
1392         ],
1393     },
1394     "x-from-ocf": [
1395         "minvalue = range[0]",
1396         "otherwise: minvalue = -MAXINT"
1397     ]
1398 },
1399 },
1400 "maxvalue": {
1401     "type": "number",
1402     "x-ocf-conversion": {
1403         "x-ocf-alias": "oic.r.temperature",
1404         "x-to-ocf": [
1405             "range[1] = maxvalue"
1406         ],
1407     },
1408     "x-from-ocf": [
1409         "maxvalue = range[1]",
1410         "otherwise: maxvalue = MAXINT"
1411     ]
1412 },
1413 "step": {
1414     "type": "number",
1415     "x-ocf-conversion": {
1416         "x-ocf-alias": "oic.r.temperature",
1417         "x-to-ocf": [
1418             "ocf.step = step"
1419         ],
1420     },
1421     "x-from-ocf": [
1422         "step = ocf.step",
1423         "otherwise: step = undefined (0x00)"

```

```

1423         ]
1424     }
1425 }
1426 }
1427 }
1428 },
1429 "type": "object",
1430 "allOf": [
1431     {"$ref": "#/definitions/asa.environment.targettemperature"}
1432 ],
1433 "required": [ "targetvalue", "minvalue", "maxvalue", "step" ]
1434 }
1435
1436 example: /
1437 {
1438     "rt":      ["oic.r.temperature"]
1439 }
1440
1441 post:
1442 body:
1443 application/json:
1444 schema: /
1445 {
1446     "id":
1447 "http://openinterconnect.org/asamapping/schemas/asa.environment.targettemperature.json#",
1448     "$schema": "http://json-schema.org/draft-04/schema#",
1449     "description": "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1450 reserved.",
1451     "title": "Target Temperature",
1452     "definitions": {
1453         "asa.environment.targettemperature": {
1454             "type": "object",
1455             "properties": {
1456                 "targetvalue": {
1457                     "type": "number",
1458                     "description": "Measured value",
1459                     "x-ocf-conversion": {
1460                         "x-ocf-alias": "oic.r.temperature",
1461                         "x-to-ocf": [
1462                             "temperature = targetvalue",
1463                             "units = C"
1464                         ],
1465                         "x-from-ocf": {
1466                             "oneOf": [
1467                                 {
1468                                     "properties": {
1469                                         "units": "string",
1470                                         "enum": ["C"]
1471                                     },
1472                                     "x-from-ocf": [
1473                                         "targetvalue = temperature"
1474                                     ]
1475                                 },
1476                                 {
1477                                     "properties": {
1478                                         "units": "string",
1479                                         "enum": ["F"]
1480                                     },
1481                                     "x-from-ocf": [
1482                                         "targetvalue = (temperature-32)*5/9"
1483                                     ]
1484                                 },
1485                                 {
1486                                     "properties": {
1487                                         "units": "string",
1488                                         "enum": ["K"]
1489                                     },
1490                                     "x-from-ocf": [

```

```

1491         "targetvalue = temperature-273.15"
1492     ]
1493     }
1494 ]
1495 }
1496 }
1497 },
1498 "minvalue": {
1499     "type": "number",
1500     "x-ocf-conversion": {
1501         "x-ocf-alias": "oic.r.temperature",
1502         "x-to-ocf": [
1503             "range[0] = minvalue"
1504         ],
1505         "x-from-ocf": [
1506             "minvalue = range[0]",
1507             "otherwise: minvalue = -MAXINT"
1508         ]
1509     }
1510 },
1511 "maxvalue": {
1512     "type": "number",
1513     "x-ocf-conversion": {
1514         "x-ocf-alias": "oic.r.temperature",
1515         "x-to-ocf": [
1516             "range[1] = maxvalue"
1517         ],
1518         "x-from-ocf": [
1519             "maxvalue = range[1]",
1520             "otherwise: maxvalue = MAXINT"
1521         ]
1522     }
1523 },
1524 "step": {
1525     "type": "number",
1526     "x-ocf-conversion": {
1527         "x-ocf-alias": "oic.r.temperature",
1528         "x-to-ocf": [
1529             "ocf.step = step"
1530         ],
1531         "x-from-ocf": [
1532             "step = ocf.step",
1533             "otherwise: step = undefined (0x00)"
1534         ]
1535     }
1536 }
1537 }
1538 }
1539 },
1540 "type": "object",
1541 "allOf": [
1542     {"$ref": "#/definitions/asa.environment.targettemperature"}
1543 ],
1544 "required": [ "targetvalue", "minvalue", "maxvalue", "step" ]
1545 }
1546
1547 responses :
1548     200:
1549         body:
1550             application/json:
1551                 schema: /
1552                     {
1553                         "id":
1554 "http://openinterconnect.org/asamapping/schemas/asa.environment.targettemperature.json#",
1555                         "$schema": "http://json-schema.org/draft-04/schema#",
1556                         "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1557 reserved.",
1558                         "title": "Target Temperature",

```

```

1559 "definitions": {
1560   "asa.environment.targettemperature": {
1561     "type": "object",
1562     "properties": {
1563       "targetvalue": {
1564         "type": "number",
1565         "description": "Measured value",
1566         "x-ocf-conversion": {
1567           "x-ocf-alias": "oic.r.temperature",
1568           "x-to-ocf": [
1569             "temperature = targetvalue",
1570             "units = C"
1571           ],
1572           "x-from-ocf": {
1573             "oneOf": [
1574               {
1575                 "properties": {
1576                   "units": "string",
1577                   "enum": ["C"]
1578                 },
1579                 "x-from-ocf": [
1580                   "targetvalue = temperature"
1581                 ]
1582               },
1583               {
1584                 "properties": {
1585                   "units": "string",
1586                   "enum": ["F"]
1587                 },
1588                 "x-from-ocf": [
1589                   "targetvalue = (temperature-32)*5/9"
1590                 ]
1591               },
1592               {
1593                 "properties": {
1594                   "units": "string",
1595                   "enum": ["K"]
1596                 },
1597                 "x-from-ocf": [
1598                   "targetvalue = temperature-273.15"
1599                 ]
1600               }
1601             ]
1602           }
1603         },
1604         "minvalue": {
1605           "type": "number",
1606           "x-ocf-conversion": {
1607             "x-ocf-alias": "oic.r.temperature",
1608             "x-to-ocf": [
1609               "range[0] = minvalue"
1610             ],
1611             "x-from-ocf": [
1612               "minvalue = range[0]",
1613               "otherwise: minvalue = -MAXINT"
1614             ]
1615           }
1616         },
1617         "maxvalue": {
1618           "type": "number",
1619           "x-ocf-conversion": {
1620             "x-ocf-alias": "oic.r.temperature",
1621             "x-to-ocf": [
1622               "range[1] = maxvalue"
1623             ],
1624             "x-from-ocf": [
1625               "maxvalue = range[1]",
1626               "otherwise: maxvalue = MAXINT"
1627             ]
1628           }
1629         }
1630       }
1631     }
1632   }
1633 }

```



```

1630     },
1631     "step": {
1632       "type": "number",
1633       "x-ocf-conversion": {
1634         "x-ocf-alias": "oic.r.temperature",
1635         "x-to-ocf": [
1636           "ocf.step = step"
1637         ],
1638         "x-from-ocf": [
1639           "step = ocf.step",
1640           "otherwise: step = undefined (0x00)"
1641         ]
1642       }
1643     }
1644   }
1645 }
1646 },
1647 "type": "object",
1648 "allOf": [
1649   {"$ref": "#/definitions/asa.environment.targettemperature"}
1650 ],
1651 "required": [ "targetvalue", "minvalue", "maxvalue", "step" ]
1652 }
1653

```

1654 8.7.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
targetvalue	oic.r.temperature	temperature = targetvalueunits = C	oneOf	Measured value
step	oic.r.temperature	ocf.step = step	step = ocf.stepotherwise: step = undefined (0x00)	
maxvalue	oic.r.temperature	range[1] = maxvalue	maxvalue = range[1]otherwise: maxvalue = MAXINT	
minvalue	oic.r.temperature	range[0] = minvalue	minvalue = range[0]otherwise: minvalue = - MAXINT	

1655 8.7.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/TargetTemperatureResURI		get	post		

1656 8.8 Audio Volume Mapping

1657 8.8.1 Introduction

1658 This API defines the mapping between an instance of an OCF Audio Controls and the AllJoyn
1659 Audio Volume interface.

1660 8.8.2 Example URI

1661 /AudioVolumeResURI

1662 8.8.3 Resource Type

1663 The resource type (rt) is defined as: oic.r.audio.

1664 8.8.4 RAML Definition

1665 #%RAML 0.8

```

1666 title: AudioVolumeInterfaceMapping
1667 version: OCFv1.0.0-20170317
1668 traits:
1669   - interface-all :
1670     queryParameters:
1671       if:
1672         enum: ["oic.if.a", "oic.if.baseline"]
1673
1674 /AudioVolumeResURI:
1675   description: |
1676     This API defines the mapping between an instance of an OCF Audio Controls
1677     and the AllJoyn Audio Volume interface.
1678
1679   is : ['interface-all']
1680   get:
1681     responses :
1682       200:
1683         body:
1684           application/json:
1685             schema: /
1686               {
1687                 "id":
1688 "http://openinterconnect.org/asamapping/schemas/asa.operation.audiovolume.json#",
1689                 "$schema": "http://json-schema.org/draft-04/schema#",
1690                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1691 reserved.",
1692                 "title": "Audio Volume",
1693                 "definitions": {
1694                   "asa.operation.audiovolume": {
1695                     "type": "object",
1696                     "properties": {
1697                       "volume": {
1698                         "type": "integer",
1699                         "description": "Speaker volume index",
1700                         "x-ocf-conversion": {
1701                           "x-ocf-alias": "oic.r.audio",
1702                           "x-to-ocf": [
1703                             "ocf.volume = volume"
1704                           ],
1705                           "x-from-ocf": [
1706                             "volume = ocf.volume"
1707                           ]
1708                         }
1709                       },
1710                       "maxvolume": {
1711                         "type": "integer",
1712                         "x-ocf-conversion": {
1713                           "x-ocf-alias": "oic.r.audio",
1714                           "x-to-ocf": [
1715                             "range[0] = 0",
1716                             "range[1] = maxvolume"
1717                           ],
1718                           "x-from-ocf": [
1719                             "maxvolume = range[1]",
1720                             "otherwise: maxvalue = 100"
1721                           ]
1722                         }
1723                       },
1724                       "mute": {
1725                         "type": "boolean",
1726                         "x-ocf-conversion": {
1727                           "x-ocf-alias": "oic.r.audio",
1728                           "x-to-ocf": [
1729                             "ocf.mute = mute"
1730                           ],

```

```

1731         "x-from-ocf": [
1732             "mute = ocf.mute"
1733         ]
1734     }
1735 }
1736 }
1737 }
1738 },
1739 "type": "object",
1740 "allOf": [
1741     {"$ref": "#/definitions/asa.operation.audiovolume"}
1742 ],
1743 "required": [ "volume", "maxvolume", "mute" ]
1744 }
1745
1746 example: /
1747     {
1748         "rt":      ["oic.r.audio"]
1749     }
1750
1751 post:
1752 body:
1753 application/json:
1754 schema: /
1755     {
1756         "id": "http://openinterconnect.org/asamapping/schemas/asa.operation.audiovolume.json#",
1757         "$schema": "http://json-schema.org/draft-04/schema#",
1758         "description": "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1759 reserved.",
1760         "title": "Audio Volume",
1761         "definitions": {
1762             "asa.operation.audiovolume": {
1763                 "type": "object",
1764                 "properties": {
1765                     "volume": {
1766                         "type": "integer",
1767                         "description": "Speaker volume index",
1768                         "x-ocf-conversion": {
1769                             "x-ocf-alias": "oic.r.audio",
1770                             "x-to-ocf": [
1771                                 "ocf.volume = volume"
1772                             ],
1773                             "x-from-ocf": [
1774                                 "volume = ocf.volume"
1775                             ]
1776                         }
1777                     },
1778                     "maxvolume": {
1779                         "type": "integer",
1780                         "x-ocf-conversion": {
1781                             "x-ocf-alias": "oic.r.audio",
1782                             "x-to-ocf": [
1783                                 "range[0] = 0",
1784                                 "range[1] = maxvolume"
1785                             ],
1786                             "x-from-ocf": [
1787                                 "maxvolume = range[1]",
1788                                 "otherwise: maxvalue = 100"
1789                             ]
1790                         }
1791                     },
1792                     "mute": {
1793                         "type": "boolean",
1794                         "x-ocf-conversion": {
1795                             "x-ocf-alias": "oic.r.audio",
1796                             "x-to-ocf": [
1797                                 "ocf.mute = mute"
1798                             ],

```

```

1799         "x-from-ocf": [
1800             "mute = ocf.mute"
1801         ]
1802     }
1803 }
1804 }
1805 }
1806 },
1807 "type": "object",
1808 "allOf": [
1809     { "$ref": "#/definitions/asa.operation.audiovolume" }
1810 ],
1811 "required": [ "volume", "maxvolume", "mute" ]
1812 }
1813
1814 responses :
1815     200:
1816         body:
1817             application/json:
1818                 schema: /
1819                     {
1820                         "id":
1821 "http://openinterconnect.org/asamapping/schemas/asa.operation.audiovolume.json#",
1822 "$schema": "http://json-schema.org/draft-04/schema#",
1823 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1824 reserved.",
1825 "title": "Audio Volume",
1826 "definitions": {
1827     "asa.operation.audiovolume": {
1828         "type": "object",
1829         "properties": {
1830             "volume": {
1831                 "type": "integer",
1832                 "description": "Speaker volume index",
1833                 "x-ocf-conversion": {
1834                     "x-ocf-alias": "oic.r.audio",
1835                     "x-to-ocf": [
1836                         "ocf.volume = volume"
1837                     ],
1838                     "x-from-ocf": [
1839                         "volume = ocf.volume"
1840                     ]
1841                 }
1842             },
1843             "maxvolume": {
1844                 "type": "integer",
1845                 "x-ocf-conversion": {
1846                     "x-ocf-alias": "oic.r.audio",
1847                     "x-to-ocf": [
1848                         "range[0] = 0",
1849                         "range[1] = maxvolume"
1850                     ],
1851                     "x-from-ocf": [
1852                         "maxvolume = range[1]",
1853                         "otherwise: maxvalue = 100"
1854                     ]
1855                 }
1856             },
1857             "mute": {
1858                 "type": "boolean",
1859                 "x-ocf-conversion": {
1860                     "x-ocf-alias": "oic.r.audio",
1861                     "x-to-ocf": [
1862                         "ocf.mute = mute"
1863                     ],
1864                     "x-from-ocf": [
1865                         "mute = ocf.mute"
1866                     ]

```

```

1867     }
1868   }
1869 }
1870 }
1871 },
1872 "type": "object",
1873 "allof": [
1874   {"$ref": "#/definitions/asa.operation.audiovolume"}
1875 ],
1876 "required": [ "volume", "maxvolume", "mute" ]
1877 }
1878

```

1879 **8.8.5 Property Definition**

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
volume	oic.r.audio	ocf.volume = volume	volume = ocf.volume	Speaker volume index
maxvolume	oic.r.audio	range[0] = Orange[1] = maxvolume	maxvolume = range[1]otherwise: maxvalue = 100	
mute	oic.r.audio	ocf.mute = mute	mute = ocf.mute	

1880 **8.8.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/AudioVolumeResURI		get	post		

1881 **8.9 Climate Control Mode Mapping**

1882 **8.9.1 Introduction**

1883 This API defines the mapping between an instance of an AllJoyn ClimateControlMode interface and the OCF equivalent Resource. ClimateControlMode has three Properties; these map as follows:
 1884 mode, supportedmodes -> Mode Resource operationalstate -> OperationalState Resource This
 1885 can be represented in OCF either as two distinct Resource instances or a single instance with two
 1886 RTs (oic.r.mode, oic.r.operationalstate)
 1887

1888 **8.9.2 Example URI**

1889 /ClimateControlModeResURI

1890 **8.9.3 Resource Type**

1891 The resource type (rt) is defined as: oic.r.mode.

1892 **8.9.4 RAML Definition**

```

1893 #%RAML 0.8
1894 title: ClimateControlModeInterfaceMapping
1895 version: OCFv1.0.0-20170317
1896 traits:
1897   - interface-actuator :
1898     queryParameters:
1899       if:
1900         enum: ["oic.if.a", "oic.if.baseline"]
1901
1902 /ClimateControlModeResURI:
1903   description: |
1904     This API defines the mapping between an instance of an AllJoyn ClimateControlMode interface and
1905     the OCF equivalent Resource.
1906     ClimateControlMode has three Properties; these map as follows:
1907     mode, supportedmodes -> Mode Resource
1908     operationalstate -> OperationalState Resource
1909     This can be represented in OCF either as two distinct Resource instances or a single instance

```

```

1910 with two RTs (oic.r.mode, oic.r.operationalstate)
1911
1912 is : ['interface-actuator']
1913 get:
1914 responses :
1915 200:
1916 body:
1917 application/json:
1918 schema: /
1919 {
1920 "id":
1921 "http://openinterconnect.org/asamapping/schemas/asa.operation.climatecontrolmode.json#",
1922 "$schema": "http://json-schema.org/draft-04/schema#",
1923 "description": "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1924 reserved.",
1925 "title": "Climate Control Mode",
1926 "definitions": {
1927 "asa.operation.climatecontrolmode": {
1928 "type": "object",
1929 "properties": {
1930 "mode": {
1931 "type": "integer",
1932 "description": "Current mode of device.",
1933 "x-ocf-conversion": {
1934 "x-ocf-alias": "oic.r.mode",
1935 "x-to-ocf": [
1936 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
1937 "ocf.mode[0] = modearray[mode]"
1938 ],
1939 "x-from-ocf": [
1940 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
1941 "mode = indexof modeArray[ocf.mode[0]]"
1942 ]
1943 }
1944 },
1945 "supportedmodes": {
1946 "type": "array",
1947 "items": {
1948 "type": "integer"
1949 },
1950 "description": "Array of supported modes",
1951 "x-ocf-conversion": {
1952 "x-ocf-alias": "oic.r.mode",
1953 "x-to-ocf": [
1954 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
1955 "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
1956 modearray[supportedmodes[x]]"
1957 ],
1958 "x-from-ocf": [
1959 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
1960 "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
1961 modearray[ocf.supportedmodes[x]]"
1962 ]
1963 }
1964 },
1965 "operationalstate": {
1966 "type": "integer",
1967 "description": "Current status of device",
1968 "x-ocf-conversion": {
1969 "x-ocf-alias": "oic.r.operationalstate",
1970 "x-to-ocf": [
1971 "machinestates =
1972 [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
1973 "currentmachinestate = machinestates[operationalstate]"
1974 ],
1975 "x-from-ocf": [
1976 "statearray =
1977 [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",

```

```

1978         "operationalstate = indexof statearray[currentmachinestate[0]]"
1979     ]
1980 }
1981 }
1982 }
1983 }
1984 },
1985 "type": "object",
1986 "allOf": [
1987   {"$ref": "#/definitions/asa.operation.climatecontrolmode"}
1988 ],
1989 "required": [ "mode","supportedmodes","operationalstate" ]
1990 }
1991
1992 example: /
1993 {
1994   "rt":      ["oic.r.mode","oic.operationalstate"]
1995 }
1996
1997 post:
1998   body:
1999     application/json:
2000     schema: /
2001       {
2002         "id":
2003         "http://openinterconnect.org/asamapping/schemas/asa.operation.climatecontrolmode.json#",
2004         "$schema": "http://json-schema.org/draft-04/schema#",
2005         "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2006 reserved.",
2007         "title": "Climate Control Mode",
2008         "definitions": {
2009           "asa.operation.climatecontrolmode": {
2010             "type": "object",
2011             "properties": {
2012               "mode": {
2013                 "type": "integer",
2014                 "description": "Current mode of device.",
2015                 "x-ocf-conversion": {
2016                   "x-ocf-alias": "oic.r.mode",
2017                   "x-to-ocf": [
2018                     "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2019                     "ocf.mode[0] = modearray[mode]"
2020                   ],
2021                   "x-from-ocf": [
2022                     "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2023                     "mode = indexof modeArray[ocf.mode[0]]"
2024                   ]
2025                 }
2026             },
2027             "supportedmodes": {
2028               "type": "array",
2029               "items": {
2030                 "type": "integer"
2031               },
2032               "description": "Array of supported modes",
2033               "x-ocf-conversion": {
2034                 "x-ocf-alias": "oic.r.mode",
2035                 "x-to-ocf": [
2036                   "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2037                   "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
2038 modearray[supportedmodes[x]]"
2039                 ],
2040                 "x-from-ocf": [
2041                   "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2042                   "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
2043 modearray[ocf.supportedmodes[x]]"
2044                 ]
2045             }

```

```

2046         },
2047         "operationalstate": {
2048             "type": "integer",
2049             "description": "Current status of device",
2050             "x-ocf-conversion": {
2051                 "x-ocf-alias": "oic.r.operationalstate",
2052                 "x-to-ocf": [
2053                     "machinestates =
2054 [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
2055                     "currentmachinestate = machinestates[operationalstate]"
2056                 ],
2057                 "x-from-ocf": [
2058                     "statearray =
2059 [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
2060                     "operationalstate = indexof statearray[currentmachinestate[0]]"
2061                 ]
2062             }
2063         }
2064     }
2065 }
2066 },
2067 "type": "object",
2068 "allOf": [
2069     { "$ref": "#/definitions/asa.operation.climatecontrolmode" }
2070 ],
2071 "required": [ "mode", "supportedmodes", "operationalstate" ]
2072 }
2073
2074 responses :
2075     200:
2076         body:
2077             application/json:
2078                 schema: /
2079                     {
2080                         "id":
2081 "http://openinterconnect.org/asamapping/schemas/asa.operation.climatecontrolmode.json#",
2082                         "$schema": "http://json-schema.org/draft-04/schema#",
2083                         "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2084 reserved.",
2085                         "title": "Climate Control Mode",
2086                         "definitions": {
2087                             "asa.operation.climatecontrolmode": {
2088                                 "type": "object",
2089                                 "properties": {
2090                                     "mode": {
2091                                         "type": "integer",
2092                                         "description": "Current mode of device.",
2093                                         "x-ocf-conversion": {
2094                                             "x-ocf-alias": "oic.r.mode",
2095                                             "x-to-ocf": [
2096                                                 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2097                                                 "ocf.mode[0] = modearray[mode]"
2098                                             ],
2099                                             "x-from-ocf": [
2100                                                 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2101                                                 "mode = indexof modeArray[ocf.mode[0]]"
2102                                             ]
2103                                         }
2104                                     },
2105                                     "supportedmodes": {
2106                                         "type": "array",
2107                                         "items": {
2108                                             "type": "integer"
2109                                         },
2110                                         "description": "Array of supported modes",
2111                                         "x-ocf-conversion": {
2112                                             "x-ocf-alias": "oic.r.mode",
2113                                             "x-to-ocf": [

```



```

2114         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2115         "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
modearray[supportedmodes[x]]"
2116     ],
2117     "x-from-ocf": [
2118         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2119         "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
modearray[ocf.supportedmodes[x]]"
2120     ],
2121     ]
2122 }
2123 },
2124 },
2125     "operationalstate": {
2126         "type": "integer",
2127         "description": "Current status of device",
2128         "x-ocf-conversion": {
2129             "x-ocf-alias": "oic.r.operationalstate",
2130             "x-to-ocf": [
2131                 "machinestates =
[Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
2132                 "currentmachinestate = machinestates[operationalstate]"
2133             ],
2134             "x-from-ocf": [
2135                 "statearray =
[Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
2136                 "operationalstate = indexof statearray[currentmachinestate[0]]"
2137             ]
2138         }
2139     }
2140 }
2141 }
2142 }
2143 }
2144 },
2145     "type": "object",
2146     "allof": [
2147         {"$ref": "#/definitions/asa.operation.climatecontrolmode"}
2148     ],
2149     "required": [ "mode", "supportedmodes", "operationalstate" ]
2150 }
2151

```

2152 8.9.5 Property Definition

['AllJ oyn'] Prope rty name	OCF Resour ce	To OCF	From OCF	Des cript ion
suppo rtedm odes	oic.r.mo de	modearray = [Off,Heat,Cool,Auto,AuxilliaryHea t,Dry,ContinuousDry]for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] = modearray[supportedmodes[x]]	modearray = [Off,Heat,Cool,Auto,AuxilliaryH eat,Dry,ContinuousDry]for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof modearray[ocf.supportedmodes [x]]	Arra y of supp orte d mod es
operat ionalst ate	oic.r.op erationa lstate	machinestates = [Idle,Heating,Cooling,PendingHe at,PendingCool,AuxilliaryHeat]cur rentmachinestate = machinestates[operationalstate]	statearray = [Idle,Heating,Cooling,PendingH eat,PendingCool,AuxilliaryHeat] operationalstate = indexof statearray[currentmachinestate[0]]	Curr ent statu s of devi ce
mode	oic.r.mo de	modearray = [Off,Heat,Cool,Auto,AuxilliaryHea t,Dry,ContinuousDry]ocf.mode[0] = modearray[mode]	modearray = [Off,Heat,Cool,Auto,AuxilliaryH eat,Dry,ContinuousDry]mode = indexof modeArray[ocf.mode[0]]	Curr ent mod e of devi ce.

2153 **8.9.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/ClimateControlModeResURI		get	post		

2154 **8.10 Closed Status Mapping**

2155 **8.10.1 Introduction**

2156 This API defines the mapping between an instance of an AllJoyn ClosedStatus Interface and the
 2157 OCF Door Resource.

2158 **8.10.2 Example URI**

2159 /ClosedStatusResURI

2160 **8.10.3 Resource Type**

2161 The resource type (rt) is defined as: oic.r.door.

2162 **8.10.4 RAML Definition**

```

2163 #%RAML 0.8
2164 title: ClosedStatusInterfaceMapping
2165 version: OCFv1.0.0-20170317
2166 traits:
2167   - interface-all :
2168     queryParameters:
2169       if:
2170         enum: ["oic.if.s", "oic.if.baseline"]
2171
2172 /ClosedStatusResURI:
2173   description: |
2174     This API defines the mapping between an instance of an AllJoyn ClosedStatus Interface and
2175     the OCF Door Resource.
2176
2177   is : ['interface-all']
2178   get:
2179     responses :
2180       200:
2181         body:
2182           application/json:
2183             schema: /
2184               {
2185                 "id":
2186 "http://openinterconnect.org/asamapping/schemas/asa.operation.closedstatus.json#",
2187                 "$schema": "http://json-schema.org/draft-04/schema#",
2188                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2189 reserved.",
2190                 "title": "Closed Status",
2191                 "definitions": {
2192                   "asa.operation.closedstatus": {
2193                     "type": "object",
2194                     "properties": {
2195                       "isclosed": {
2196                         "type": "boolean",
2197                         "description": "Open/Closed status Indicator",
2198                         "x-ocf-conversion": {
2199                           "x-ocf-alias": "oic.r.door",
2200                           "x-to-ocf": [
2201                             "if isClosed ocf.openState = Closed.",
2202                             "if !isClosed ocf.openState = Open."
2203                           ],
2204                           "x-from-ocf": [
2205                             "isClosed = (openState == Closed)"

```

```

2206     ]
2207   }
2208 }
2209 }
2210 },
2211 },
2212 "type": "object",
2213 "allOf": [
2214   {"$ref": "#/definitions/asa.operation.closedstatus"}
2215 ],
2216 "required": [ "isclosed" ]
2217 }
2218
2219 example: /
2220 {
2221   "rt":      ["oic.r.door"]
2222 }
2223

```

2224 8.10.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
isclosed	oic.r.door	if isClosed ocf.openState = Closed.if !isClosed ocf.openState = Open.	isClosed = (openState == Closed)	Open/Closed status Indicator

2225 8.10.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/ClosedStatusResURI		get			

2226 8.11 Cycle Control Mapping

2227 8.11.1 Introduction

2228 This API defines the mapping between an instance of an AllJoyn CycleControl interface and the
 2229 OCF OperationalState Resource. The AllJoyn interface also supports a Method,
 2230 ExecuteOperationalCommand; this is handled in OCF using an instance of oic.r.actuator within an
 2231 oic.r.action collection. Please see Section 8 of the Mapping Specification for specifics.

2232 8.11.2 Example URI

2233 /CycleControlResURI

2234 8.11.3 Resource Type

2235 The resource type (rt) is defined as: oic.r.operationalstate.

2236 8.11.4 RAML Definition

```

2237 #%RAML 0.8
2238 title: CycleControlInterfaceMapping
2239 version: OCFv1.0.0-20170317
2240 traits:
2241   - interface-sensor :
2242     queryParameters:
2243       if:
2244         enum: ["oic.if.s", "oic.if.baseline"]
2245
2246 /CycleControlResURI:
2247   description: |
2248     This API defines the mapping between an instance of an AllJoyn CycleControl interface and the
2249     OCF OperationalState Resource.

```

2250 The AllJoyn interface also supports a Method, ExecuteOperationalCommand; this is handled in OCF
 2251 using an instance of oic.r.actuator within an oic.r.action collection.
 2252 Please see Section 8 of the Mapping Specification for specifics.
 2253

```

2254 is : ['interface-sensor']
2255 get:
2256   responses :
2257     200:
2258       body:
2259         application/json:
2260           schema: /
2261             {
2262               "id":
2263 "http://openinterconnect.org/asamapping/schemas/asa.operation.ovencyclephase.json#",
2264               "$schema": "http://json-schema.org/draft-04/schema#",
2265               "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2266 reserved.",
2267               "title": "Oven Cycle Phase",
2268               "definitions": {
2269                 "asa.operation.ovencyclephase": {
2270                   "type": "object",
2271                   "properties": {
2272                     "cyclephase": {
2273                       "type": "integer",
2274                       "description": "Current phase of the operational cycle",
2275                       "x-ocf-conversion": {
2276                         "x-ocf-alias": "oic.r.operationalstate",
2277                         "x-to-ocf": [
2278                           "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
2279                           "currentmachinestate = phasearray[cyclephase]"
2280                         ],
2281                         "x-from-ocf": [
2282                           "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
2283                           "cyclephase = indexof statearray[currentmachinestate[0]]"
2284                         ]
2285                       }
2286                     },
2287                     "supportedcyclephases": {
2288                       "type": "array",
2289                       "items": {
2290                         "type": "integer"
2291                       },
2292                       "description": "Array of cycle phases supported by the Appliance.",
2293                       "x-ocf-conversion": {
2294                         "x-ocf-alias": "oic.r.operationalstate",
2295                         "x-to-ocf": [
2296                           "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
2297                           "for x=0, x < sizeof(supportedcyclephases): machinestates[x] =
2298 phasearray[supportedcyclephases[x]]"
2299                         ],
2300                         "x-from-ocf": [
2301                           "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
2302                           "for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof
2303 phasearray[machinestates[x]]"
2304                         ]
2305                       }
2306                     },
2307                     "getvendorphasesdescription": {
2308                       "x-ocf-type": "method",
2309                       "description": "Get cycle phases description",
2310                       "x-ocf-conversion": {
2311                         "x-ocf-alias": "oic.r.action"
2312                       }
2313                     }
2314                   }
2315                 }
2316             },
2317             "type": "object",

```

```

2318     "allOf": [
2319         { "$ref": "#/definitions/asa.operation.ovencyclephase" }
2320     ],
2321     "required": [ "cyclephase", "supportedcyclephases" ]
2322 }
2323
2324     example: /
2325     {
2326         "rt":      ["oic.r.operationalstate"]
2327     }
2328

```

2329 **8.11.5 Property Definition**

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
supportedcyclephases	oic.r.operationalstate	phasearray = [Unavailable,Preheating,Cooking,Cleaning]for x=0, x < sizeof(supportedcyclephases): machinestates[x] = phasearray[supportedcyclephases[x]]	phasearray = [Unavailable,Preheating,Cooking,Cleaning]for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof phasearray[machinestates[x]]	Array of cycle phases supported by the Appliance.
cyclephase	oic.r.operationalstate	phasearray = [Unavailable,Preheating,Cooking,Cleaning]currentmachinestate = phasearray[cyclephase]	phasearray = [Unavailable,Preheating,Cooking,Cleaning]cyclephase = indexof statearray[currentmachinestate[0]]	Current phase of the operational cycle
getvendorphasedescription	oic.r.action			Get cycle phases description

2330 **8.11.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/CycleControlResURI		get			

2331 **8.12 Fan Speed Level Mapping**

2332 **8.12.1 Introduction**

2333 This API defines the mapping between an instance of an AllJoyn FanSpeedLevel interface and an
2334 OCF AirFlow Resource. Note that the setting of the FanSpeedLevel to '0x00' (off) is handled via
2335 the 'OffControl' interface rather than writing directly to this interface. In such a case an instance of
2336 Binary Switch shall be exposed on the OCF side; this can be modeled as AirFlowControl which is
2337 then a collection of Binary Switch and AirFlow.

2338 **8.12.2 Example URI**

2339 /FanSpeedLevelResURI

2340 8.12.3 Resource Type

2341 The resource type (rt) is defined as: oic.r.airflow.

2342 8.12.4 RAML Definition

2343 `##RAML 0.8`

2344 `title: FanSpeedLevelInterfaceMapping`

2345 `version: OCFv1.0.0-20170317`

2346 `traits:`

2347 `- interface-actuator :`

2348 `queryParameters:`

2349 `if:`

2350 `enum: ["oic.if.a", "oic.if.baseline"]`

2351

2352 `/FanSpeedLevelResURI:`

2353 `description: |`

2354 `This API defines the mapping between an instance of an AllJoyn FanSpeedLevel interface and an`

2355 `OCF AirFlow Resource.`

2356 `Note that the setting of the FanSpeedLevel to '0x00' (off) is handled via the 'OffControl'`

2357 `interface rather than writing directly to this interface.`

2358 `In such a case an instance of Binary Switch shall be exposed on the OCF side; this can be`

2359 `modeled as AirFlowControl which is then a collection of Binary Switch and AirFlow.`

2360

2361 `is: ['interface-actuator']`

2362 `get:`

2363 `responses:`

2364 `200:`

2365 `body:`

2366 `application/json:`

2367 `schema: |`

2368 `{`

2369 `"id":`

2370 `"http://openinterconnect.org/asamapping/schemas/asa.operation.fanspeedlevel.json#",`

2371 `"$schema": "http://json-schema.org/draft-04/schema#",`

2372 `"description": "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights`

2373 `reserved.",`

2374 `"title": "Fan Speed Level",`

2375 `"definitions": {`

2376 `"asa.operation.fanspeedlevel": {`

2377 `"type": "object",`

2378 `"properties": {`

2379 `"fanspeedlevel": {`

2380 `"type": "integer",`

2381 `"description": "Fan speed level. 0 = off.",`

2382 `"x-ocf-conversion": {`

2383 `"x-ocf-alias": "oic.r.airflow",`

2384 `"x-to-ocf": [`

2385 `"speed = fanspeedlevel"`

2386 `],`

2387 `"x-from-ocf": [`

2388 `"fanspeedlevel = speed"`

2389 `]`

2390 `}`

2391 `},`

2392 `"maxfanspeedlevel": {`

2393 `"type": "integer",`

2394 `"description": "Max level allowed for fan speed",`

2395 `"x-ocf-conversion": {`

2396 `"x-ocf-alias": "oic.r.airflow",`

2397 `"x-to-ocf": [`

2398 `"range[0] = 0",`

2399 `"range[1] = maxfanspeedlevel"`

2400 `],`

2401 `"x-from-ocf": [`

```

2402         "maxfanspeedlevel = range[1]",
2403         "otherwise: maxfanspeedlevel = 100"
2404     ]
2405 }
2406 },
2407 "automode": {
2408     "type": "integer",
2409     "description": "Auto mode status.",
2410     "x-ocf-conversion": {
2411         "x-ocf-alias": "oic.r.airflow",
2412         "x-to-ocf": [
2413             "if automode != NotSupported(0xFF)",
2414             " ocf.automode = automode",
2415             "else no mapping"
2416         ],
2417         "x-from-ocf": [
2418             "automode = ocf.automode",
2419             "otherwise: automode = NotSupported(0xFF)"
2420         ]
2421     }
2422 }
2423 }
2424 },
2425 },
2426 "type": "object",
2427 "allOf": [
2428     {"$ref": "#/definitions/asa.operation.fanspeedlevel"}
2429 ],
2430 "required": [ "fanspeedlevel", "maxfanspeedlevel", "automode" ]
2431 }
2432
2433 example: /
2434 {
2435     "rt":      ["oic.r.airflow"]
2436 }
2437
2438 post:
2439 body:
2440 application/json:
2441 schema: /
2442 {
2443     "id":
2444 "http://openinterconnect.org/asamapping/schemas/asa.operation.fanspeedlevel.json#",
2445     "$schema": "http://json-schema.org/draft-04/schema#",
2446     "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2447 reserved.",
2448     "title": "Fan Speed Level",
2449     "definitions": {
2450         "asa.operation.fanspeedlevel": {
2451             "type": "object",
2452             "properties": {
2453                 "fanspeedlevel": {
2454                     "type": "integer",
2455                     "description": "Fan speed level. 0 = off.",
2456                     "x-ocf-conversion": {
2457                         "x-ocf-alias": "oic.r.airflow",
2458                         "x-to-ocf": [
2459                             "speed = fanspeedlevel"
2460                         ],
2461                         "x-from-ocf": [
2462                             "fanspeedlevel = speed"
2463                         ]
2464                     }
2465                 },
2466                 "maxfanspeedlevel": {
2467                     "type": "integer",
2468                     "description": "Max level allowed for fan speed",
2469                     "x-ocf-conversion": {

```

```

2470         "x-ocf-alias": "oic.r.airflow",
2471         "x-to-ocf": [
2472             "range[0] = 0",
2473             "range[1] = maxfanspeedlevel"
2474         ],
2475         "x-from-ocf": [
2476             "maxfanspeedlevel = range[1]",
2477             "otherwise: maxfanspeedlevel = 100"
2478         ]
2479     }
2480 },
2481     "automode": {
2482         "type": "integer",
2483         "description": "Auto mode status.",
2484         "x-ocf-conversion": {
2485             "x-ocf-alias": "oic.r.airflow",
2486             "x-to-ocf": [
2487                 "if automode != NotSupported(0xFF)",
2488                 "  ocf.automode = automode",
2489                 "else no mapping"
2490             ],
2491             "x-from-ocf": [
2492                 "automode = ocf.automode",
2493                 "otherwise: automode = NotSupported(0xFF)"
2494             ]
2495         }
2496     }
2497 }
2498 }
2499 },
2500 "type": "object",
2501 "allOf": [
2502     {"$ref": "#/definitions/asa.operation.fanspeedlevel"}
2503 ],
2504 "required": [ "fanspeedlevel", "maxfanspeedlevel", "automode" ]
2505 }
2506
2507 responses :
2508     200:
2509         body:
2510             application/json:
2511                 schema: /
2512                     {
2513                         "id":
2514 "http://openinterconnect.org/asamapping/schemas/asa.operation.fanspeedlevel.json#",
2515                         "$schema": "http://json-schema.org/draft-04/schema#",
2516                         "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2517 reserved.",
2518                         "title": "Fan Speed Level",
2519                         "definitions": {
2520                             "asa.operation.fanspeedlevel": {
2521                                 "type": "object",
2522                                 "properties": {
2523                                     "fanspeedlevel": {
2524                                         "type": "integer",
2525                                         "description": "Fan speed level. 0 = off.",
2526                                         "x-ocf-conversion": {
2527                                             "x-ocf-alias": "oic.r.airflow",
2528                                             "x-to-ocf": [
2529                                                 "speed = fanspeedlevel"
2530                                             ],
2531                                             "x-from-ocf": [
2532                                                 "fanspeedlevel = speed"
2533                                             ]
2534                                         }
2535                                     },
2536                                     "maxfanspeedlevel": {
2537                                         "type": "integer",

```



```

2538         "description": "Max level allowed for fan speed",
2539         "x-ocf-conversion": {
2540             "x-ocf-alias": "oic.r.airflow",
2541             "x-to-ocf": [
2542                 "range[0] = 0",
2543                 "range[1] = maxfanspeedlevel"
2544             ],
2545             "x-from-ocf": [
2546                 "maxfanspeedlevel = range[1]",
2547                 "otherwise: maxfanspeedlevel = 100"
2548             ]
2549         }
2550     },
2551     "automode": {
2552         "type": "integer",
2553         "description": "Auto mode status.",
2554         "x-ocf-conversion": {
2555             "x-ocf-alias": "oic.r.airflow",
2556             "x-to-ocf": [
2557                 "if automode != NotSupported(0xFF)",
2558                 " ocf.automode = automode",
2559                 "else no mapping"
2560             ],
2561             "x-from-ocf": [
2562                 "automode = ocf.automode",
2563                 "otherwise: automode = NotSupported(0xFF)"
2564             ]
2565         }
2566     }
2567 }
2568 },
2569 "type": "object",
2570 "allOf": [
2571     {"$ref": "#/definitions/asa.operation.fanspeedlevel"}
2572 ],
2573 "required": [ "fanspeedlevel", "maxfanspeedlevel", "automode" ]
2574 }
2575 }
2576 }

```

2577 **8.12.5 Property Definition**

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
maxfanspeedlevel	oic.r.airflow	range[0] = 0 range[1] = maxfanspeedlevel	maxfanspeedlevel = range[1] otherwise: maxfanspeedlevel = 100	Max level allowed for fan speed
automode	oic.r.airflow	if automode != NotSupported(0xFF)) ocf.automode = automode else no mapping	automode = ocf.automode otherwise: automode = NotSupported(0xFF)	Auto mode status.
fanspeedlevel	oic.r.airflow	speed = fanspeedlevel	fanspeedlevel = speed	Fan speed level. 0 = off.

2578 **8.12.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/FanSpeedLevelResURI		get	post		

2579 **8.13 Heating Zone Mapping**

2580 **8.13.1 Introduction**

2581 This API defines the mapping between an instance of an AllJoyn HeatingZone interface and an
2582 OCF HeatingZoneCollection Resource. Each element in the array of heating zones within the

2583 AllJoyn HeatingZone interface maps to an instance of OCF HeatingZone, itself a link in an instance
2584 of an OCF HeatingZoneCollection. The mapping defined in the schema describes the population
2585 of the OCF HeatingZone Resource that constitutes the Resources that are contained in the
2586 collection.

2587 **8.13.2 Example URI**

2588 /HeatingZoneResURI

2589 **8.13.3 Resource Type**

2590 The resource type (rt) is defined as: oic.r.heatingzonecollection.

2591 **8.13.4 RAML Definition**

```
2592 #%RAML 0.8
2593 title: HeatingZoneInterfaceMapping
2594 version: OCFv1.0.0-20170317
2595 traits:
2596   - interface-sensor :
2597     queryParameters:
2598       if:
2599         enum: ["oic.if.s", "oic.if.baseline"]
2600
2601 /HeatingZoneResURI:
2602   description: |
2603     This API defines the mapping between an instance of an AllJoyn HeatingZone interface and an OCF
2604     HeatingZoneCollection Resource.
2605     Each element in the array of heating zones within the AllJoyn HeatingZone interface maps to an
2606     instance of OCF HeatingZone, itself a link in an instance of an OCF HeatingZoneCollection.
2607     The mapping defined in the schema describes the population of the OCF HeatingZone Resource that
2608     constitutes the Resources that are contained in the collection.
2609
2610   is : ['interface-sensor']
2611   get:
2612     responses :
2613       200:
2614         body:
2615           application/json:
2616             schema: /
2617               {
2618                 "id":
2619 "http://openinterconnect.org/asamapping/schemas/asa.operation.heatingzone.json#",
2620                 "$schema": "http://json-schema.org/draft-04/schema#",
2621                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2622 reserved.",
2623                 "title": "Heating Zone",
2624                 "definitions": {
2625                   "asa.operation.heatingzone": {
2626                     "type": "object",
2627                     "properties": {
2628                       "numberofheatingzones": {
2629                         "type": "integer",
2630                         "description": "Number of heating zones.",
2631                         "x-ocf-conversion": {
2632                           "x-ocf-alias": "oic.r.heatingzonecollection",
2633                           "x-to-ocf": [
2634                             "number of links in the collection = numberofheatingzones"
2635                           ],
2636                           "x-from-ocf": [
2637                             "numberofheatingzones = number of links in the collection"
2638                           ]
2639                         }
2640                     }
2641                   }
2642                 }
2643               }
```

```

2641         "maxheatinglevels": {
2642             "type": "array",
2643             "items": {
2644                 "type": "integer"
2645             },
2646             "description": "Max heating levels for each zone",
2647             "x-ocf-conversion": {
2648                 "x-ocf-alias": "oic.r.heatingzone",
2649                 "x-to-ocf": [
2650                     "Instance of oic.r.heatingzone per array item ",
2651                     "for x=0, x<sizeof(maxheatinglevels): ocf.maxheatinglevel =
2652 maxheatinglevels[x]"
2653                 ],
2654                 "x-from-ocf": [
2655                     "for x=0;x<numlinks(oic.r.heatingzonecollection): maxheatinglevels[x] =
2656 ocf.maxheatinglevel"
2657                 ]
2658             },
2659             "heatinglevels": {
2660                 "type": "array",
2661                 "items": {
2662                     "type": "integer"
2663                 },
2664                 "description": "Current heating levels for each zone.",
2665                 "x-ocf-conversion": {
2666                     "x-ocf-alias": "oic.r.heatingzone",
2667                     "x-to-ocf": [
2668                         "Instance of oic.r.heatingzone per array item ",
2669                         "for x=0, x<sizeof(heatinglevels): ocf.heatinglevel =
2670 maxheatinglevels[x]"
2671                     ],
2672                     "x-from-ocf": [
2673                         "for x=0;x<numlinks(oic.r.heatingzonecollection): heatinglevels[x] =
2674 ocf.heatinglevel"
2675                     ]
2676                 },
2677             },
2678         },
2679     },
2680 },
2681 },
2682 "type": "object",
2683 "allOf": [
2684     { "$ref": "#/definitions/asa.operation.heatingzone" }
2685 ],
2686 "required": [ "numberofheatingzones", "maxheatinglevels", "heatinglevels" ]
2687 }
2688
2689 example: /
2690 {
2691     "rt": ["oic.r.heatingzonecollection"]
2692 }
2693

```

2694 8.13.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Descri ption
heatinglevels	oic.r.heatingzone	Instance of oic.r.heatingzone per array item for x=0, x<sizeof(heatinglevels): ocf.heatinglevel =	for x=0;x<numlinks(oic.r.heatingzonecollection): heatinglevels[x] = ocf.heatinglevel	Current heating levels for each zone.

		maxheatinglevels [x]		
numberofheatingzones	oic.r.heatingzonecollection	number of links in the collection = numberofheatingzones	numberofheatingzones = number of links in the collection	Number of heating zones.
maxheatinglevels	oic.r.heatingzone	Instance of oic.r.heatingzone per array item for x=0, x<sizeof(maxheatinglevels): ocf.maxheatinglevel = maxheatinglevels [x]	for x=0;x<numlinks(oic.r.heatingzonecollection): maxheatinglevels[x] = ocf.maxheatinglevel	Max heating levels for each zone

2695 **8.13.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/HeatingZoneResURI		get			

2696 **8.14 HVAC Fan Mode Mapping**

2697 **8.14.1 Introduction**

2698 This API defines the mapping between an instance of an AllJoyn HvacFanMode interface and an
2699 OCF Mode Resource.

2700 **8.14.2 Example URI**

2701 /HvacFanModeResURI

2702 **8.14.3 Resource Type**

2703 The resource type (rt) is defined as: oic.r.mode.

2704 **8.14.4 RAML Definition**

```

2705 #%RAML 0.8
2706 title: HvacFanModeInterfaceMapping
2707 version: OCFv1.0.0-20170317
2708 traits:
2709   - interface-actuator :
2710     queryParameters:
2711       if:
2712         enum: ["oic.if.a", "oic.if.baseline"]
2713
2714 /HvacFanModeResURI:
2715   description: |
2716     This API defines the mapping between an instance of an AllJoyn HvacFanMode interface and an OCF
2717     Mode Resource.
2718
2719   is : ['interface-actuator']
2720   get:
2721     responses :
2722       200:
2723         body:
2724           application/json:
2725             schema: /

```

```

2726     {
2727         "id":
2728 "http://openinterconnect.org/asamapping/schemas/asa.operation.hvacfanmode.json#",
2729         "$schema": "http://json-schema.org/draft-04/schema#",
2730         "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2731 reserved.",
2732         "title": "HVAC Fan Mode",
2733         "definitions": {
2734             "asa.operation.hvacfanmode": {
2735                 "type": "object",
2736                 "properties": {
2737                     "mode": {
2738                         "type": "integer",
2739                         "description": "Current mode of device.",
2740                         "x-ocf-conversion": {
2741                             "x-ocf-alias": "oic.r.mode",
2742                             "x-to-ocf": [
2743                                 "modearray = [Auto,Circulation,Continuous]",
2744                                 "ocf.mode[0] = modearray[mode]"
2745                             ],
2746                             "x-from-ocf": [
2747                                 "modearray = [Auto,Circulation,Continuous]",
2748                                 "mode = indexof modeArray[ocf.mode[0]]"
2749                             ]
2750                         }
2751                     },
2752                     "supportedmodes": {
2753                         "type": "array",
2754                         "items": {
2755                             "type": "integer"
2756                         },
2757                         "description": "Array of supported modes",
2758                         "x-ocf-conversion": {
2759                             "x-ocf-alias": "oic.r.mode",
2760                             "x-to-ocf": [
2761                                 "modearray = [Auto,Circulation,Continuous]",
2762                                 "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
2763 modearray[supportedmodes[x]]"
2764                             ],
2765                             "x-from-ocf": [
2766                                 "modearray = [Auto,Circulation,Continuous]",
2767                                 "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
2768 modearray[ocf.supportedmodes[x]]"
2769                             ]
2770                         }
2771                     }
2772                 }
2773             }
2774         },
2775         "type": "object",
2776         "allOf": [
2777             {"$ref": "#/definitions/asa.operation.hvacfanmode"}
2778         ],
2779         "required": [ "mode", "supportedmodes" ]
2780     }
2781
2782     example: /
2783     {
2784         "rt":      ["oic.r.mode"]
2785     }
2786
2787     post:
2788     body:
2789     application/json:
2790     schema: /
2791     {
2792         "id": "http://openinterconnect.org/asamapping/schemas/asa.operation.hvacfanmode.json#",
2793         "$schema": "http://json-schema.org/draft-04/schema#",

```

```

2794         "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2795 reserved.",
2796         "title": "HVAC Fan Mode",
2797         "definitions": {
2798             "asa.operation.hvacfanmode": {
2799                 "type": "object",
2800                 "properties": {
2801                     "mode": {
2802                         "type": "integer",
2803                         "description": "Current mode of device.",
2804                         "x-ocf-conversion": {
2805                             "x-ocf-alias": "oic.r.mode",
2806                             "x-to-ocf": [
2807                                 "modearray = [Auto,Circulation,Continuous]",
2808                                 "ocf.mode[0] = modearray[mode]"
2809                             ],
2810                             "x-from-ocf": [
2811                                 "modearray = [Auto,Circulation,Continuous]",
2812                                 "mode = indexof modeArray[ocf.mode[0]]"
2813                             ]
2814                         }
2815                     },
2816                     "supportedmodes": {
2817                         "type": "array",
2818                         "items": {
2819                             "type": "integer"
2820                         },
2821                         "description": "Array of supported modes",
2822                         "x-ocf-conversion": {
2823                             "x-ocf-alias": "oic.r.mode",
2824                             "x-to-ocf": [
2825                                 "modearray = [Auto,Circulation,Continuous]",
2826                                 "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
2827 modearray[supportedmodes[x]]"
2828                             ],
2829                             "x-from-ocf": [
2830                                 "modearray = [Auto,Circulation,Continuous]",
2831                                 "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
2832 modearray[ocf.supportedmodes[x]]"
2833                             ]
2834                         }
2835                     }
2836                 }
2837             }
2838         },
2839         "type": "object",
2840         "allOf": [
2841             {"$ref": "#/definitions/asa.operation.hvacfanmode"}
2842         ],
2843         "required": [ "mode", "supportedmodes" ]
2844     }
2845
2846     responses :
2847         200:
2848             body:
2849                 application/json:
2850                     schema: /
2851                         {
2852                             "id":
2853 "http://openinterconnect.org/asamapping/schemas/asa.operation.hvacfanmode.json#",
2854                             "$schema": "http://json-schema.org/draft-04/schema#",
2855                             "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2856 reserved.",
2857                             "title": "HVAC Fan Mode",
2858                             "definitions": {
2859                                 "asa.operation.hvacfanmode": {
2860                                     "type": "object",
2861                                     "properties": {

```

```

2862         "mode": {
2863             "type": "integer",
2864             "description": "Current mode of device.",
2865             "x-ocf-conversion": {
2866                 "x-ocf-alias": "oic.r.mode",
2867                 "x-to-ocf": [
2868                     "modearray = [Auto,Circulation,Continuous]",
2869                     "ocf.mode[0] = modearray[mode]"
2870                 ],
2871                 "x-from-ocf": [
2872                     "modearray = [Auto,Circulation,Continuous]",
2873                     "mode = indexof modeArray[ocf.mode[0]]"
2874                 ]
2875             }
2876         },
2877         "supportedmodes": {
2878             "type": "array",
2879             "items": {
2880                 "type": "integer"
2881             },
2882             "description": "Array of supported modes",
2883             "x-ocf-conversion": {
2884                 "x-ocf-alias": "oic.r.mode",
2885                 "x-to-ocf": [
2886                     "modearray = [Auto,Circulation,Continuous]",
2887                     "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
2888 modearray[supportedmodes[x]]"
2889                 ],
2890                 "x-from-ocf": [
2891                     "modearray = [Auto,Circulation,Continuous]",
2892                     "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
2893 modearray[ocf.supportedmodes[x]]"
2894                 ]
2895             }
2896         }
2897     }
2898 },
2899 "type": "object",
2900 "allOf": [
2901     {"$ref": "#/definitions/asa.operation.hvacfanmode"}
2902 ],
2903 "required": [ "mode", "supportedmodes" ]
2904 }
2905 }
2906

```

2907 8.14.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Descript ion
supportedm odes	oic.r.m ode	modearray [Auto,Circulation,Continuous]f or x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] = modearray[supportedmodes[x]]	modearray [Auto,Circulation,Continu ous]for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof modearray[ocf.supported modes[x]]	Array of supporte d modes
mode	oic.r.m ode	modearray [Auto,Circulation,Continuous]o cf.mode[0] = modearray[mode]	modearray [Auto,Circulation,Continu ous]mode = indexof modeArray[ocf.mode[0]]	Current mode of device.

2908 8.14.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
----------	--------	------	--------	--------	--------

/HvacFanModeResURI		get	post		
--------------------	--	-----	------	--	--

2909 **8.15 On Off Mapping**

2910 **8.15.1 Introduction**

2911 This API defines the mapping between an instance of an OCF Binary Switch Resource and the
 2912 equivalent Interface set by AllJoyn. A discovered instance of a Binary Switch is always mapped
 2913 to an Operation.OnOffStatus interface. A RETRIEVE on a Binary Switch maps to an action on an
 2914 instance of an Operation.OnOffStatus Interface. An UPDATE on a Binary Switch maps to a method
 2915 invocation on either Operation.OnControl or OffControl. value = true maps to Operation.OnControl
 2916 value = false maps to Operation.OffControl

2917 **8.15.2 Example URI**

2918 /OnOffResURI

2919 **8.15.3 Resource Type**

2920 The resource type (rt) is defined as: oic.r.switch.binary.

2921 **8.15.4 RAML Definition**

```

2922 #RAML 0.8
2923 title: OnOffInterfaceMapping
2924 version: OCFv1.0.0-20170317
2925 traits:
2926   - interface-actuator :
2927     queryParameters:
2928       if:
2929         enum: ["oic.if.a", "oic.if.baseline"]
2930   - interface-all :
2931     queryParameters:
2932       if:
2933         enum: ["oic.if.s", "oic.if.a", "oic.if.baseline"]
2934
2935 /OnOffResURI:
2936   description: |
2937     This API defines the mapping between an instance of an OCF Binary Switch Resource and the
2938     equivalent Interface set by AllJoyn
2939     A discovered instance of a Binary Switch is always
2940     mapped to an Operation.OnOffStatus interface.
2941     A RETRIEVE on a Binary Switch maps to an action on an instance of an Operation.OnOffStatus
2942     Interface.
2943     An UPDATE on a Binary Switch maps to a method invocation on either Operation.OnControl or
2944     OffControl.
2945     value = true maps to Operation.OnControl
2946     value = false maps to Operation.OffControl
2947
2948   get:
2949     is : ['interface-all']
2950     responses :
2951       200:
2952         body:
2953           application/json:
2954             schema: /
2955               {
2956                 "id":
2957 "http://openinterconnect.org/asamapping/schemas/asa.operation.onoffstatus.json#",
2958                 "$schema": "http://json-schema.org/draft-04/schema#",
2959                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2960 reserved.",
2961                 "title": "On Off Mapping",

```



```

2962     "definitions": {
2963         "asa.operation.onoffstatus": {
2964             "type": "object",
2965             "properties": {
2966                 "onoff": {
2967                     "type": "boolean",
2968                     "description": "On/Off status of the device",
2969                     "x-ocf-conversion": {
2970                         "x-ocf-alias": "oic.r.switch.binary",
2971                         "x-to-ocf": [
2972                             "value = onoff"
2973                         ],
2974                         "x-from-ocf": [
2975                             "onoff = value"
2976                         ]
2977                     }
2978                 }
2979             }
2980         },
2981     },
2982     "type": "object",
2983     "allOf": [
2984         { "$ref": "#/definitions/asa.operation.onoffstatus" }
2985     ],
2986     "required": [ "onoff" ]
2987 }
2988
2989     example: /
2990     {
2991         "rt":      ["oic.r.switch.binary"]
2992     }
2993
2994     post:
2995     is : ['interface-actuator']
2996     body:
2997     application/json:
2998     schema: /
2999     {
3000         "id": "http://openinterconnect.org/asamapping/schemas/asa.operation.oncontrol.json#",
3001         "$schema": "http://json-schema.org/draft-04/schema#",
3002         "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
3003 reserved.",
3004         "title": "On/Off Control",
3005         "definitions": {
3006             "oic.r.switch.binary": {
3007                 "properties": {
3008                     "oneOf": [
3009                         {
3010                             "properties": {
3011                                 "value": {
3012                                     "type": "boolean",
3013                                     "enum": [true]
3014                                 },
3015                                 "x-ocf-conversion": {
3016                                     "x-x-from-ocf": [
3017                                         "asa.operation.oncontrol::switchon()"
3018                                     ]
3019                                 }
3020                             }
3021                         },
3022                     {
3023                         "properties": {
3024                             "value": {
3025                                 "type": "boolean",
3026                                 "enum": [false]
3027                             },
3028                             "x-ocf-conversion": {

```

```

3029         "x-x-from-ocf": [
3030             "asa.operation.offcontrol::switchoff()"
3031         ]
3032     }
3033 }
3034 }
3035 ]
3036 }
3037 },
3038 "asa.operation.oncontrol": {
3039     "type": "object",
3040     "properties": {
3041         "switchon": {
3042             "type": "string",
3043             "format": "method",
3044             "description": "Turn on the device",
3045             "x-ocf-conversion": {
3046                 "x-ocf-alias": "oic.r.switch.binary",
3047                 "x-to-ocf": [
3048                     "value = true"
3049                 ]
3050             }
3051         }
3052     }
3053 },
3054 "asa.operation.offcontrol": {
3055     "type": "object",
3056     "properties": {
3057         "switchon": {
3058             "type": "string",
3059             "format": "method",
3060             "description": "Turn off the device",
3061             "x-ocf-conversion": {
3062                 "x-ocf-alias": "oic.r.switch.binary",
3063                 "x-to-ocf": [
3064                     "value = false"
3065                 ]
3066             }
3067         }
3068     }
3069 },
3070 },
3071 "type": "object",
3072 "oneOf": [
3073     {"$ref": "#/definitions/oic.r.switch.binary"},
3074     {"$ref": "#/definitions/asa.operation.oncontrol"},
3075     {"$ref": "#/definitions/asa.operation.offcontrol"}
3076 ]
3077 }
3078
3079 responses :
3080 200:
3081     body:
3082         application/json:
3083             schema: /
3084                 {
3085                     "id":
3086 "http://openinterconnect.org/asamapping/schemas/asa.operation.oncontrol.json#",
3087                     "$schema": "http://json-schema.org/draft-04/schema#",
3088                     "description": "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
3089 reserved.",
3090                     "title": "On/Off Control",
3091                     "definitions": {
3092                         "oic.r.switch.binary": {
3093                             "properties": {
3094                                 "oneOf": [
3095                                     {
3096                                         "properties": {

```

```

3097         "value": {
3098             "type": "boolean",
3099             "enum": [true]
3100         },
3101         "x-ocf-conversion": {
3102             "x-x-from-ocf": [
3103                 "asa.operation.oncontrol::switchon()"
3104             ]
3105         }
3106     },
3107     {
3108         "properties": {
3109             "value": {
3110                 "type": "boolean",
3111                 "enum": [false]
3112             },
3113             "x-ocf-conversion": {
3114                 "x-x-from-ocf": [
3115                     "asa.operation.offcontrol::switchoff()"
3116                 ]
3117             }
3118         }
3119     }
3120 ]
3121 },
3122 ],
3123 },
3124 "asa.operation.oncontrol": {
3125     "type": "object",
3126     "properties": {
3127         "switchon": {
3128             "type": "string",
3129             "format": "method",
3130             "description": "Turn on the device",
3131             "x-ocf-conversion": {
3132                 "x-ocf-alias": "oic.r.switch.binary",
3133                 "x-to-ocf": [
3134                     "value = true"
3135                 ]
3136             }
3137         }
3138     }
3139 },
3140 "asa.operation.offcontrol": {
3141     "type": "object",
3142     "properties": {
3143         "switchon": {
3144             "type": "string",
3145             "format": "method",
3146             "description": "Turn off the device",
3147             "x-ocf-conversion": {
3148                 "x-ocf-alias": "oic.r.switch.binary",
3149                 "x-to-ocf": [
3150                     "value = false"
3151                 ]
3152             }
3153         }
3154     }
3155 },
3156 },
3157 "type": "object",
3158 "oneOf": [
3159     {"$ref": "#/definitions/oic.r.switch.binary"},
3160     {"$ref": "#/definitions/asa.operation.oncontrol"},
3161     {"$ref": "#/definitions/asa.operation.offcontrol"}
3162 ]
3163 }
3164

```

3165 **8.15.5 Property Definition**

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
onoff	oic.r.switch.binary	value = onoff	onoff = value	On/Off status of the device

3166 **8.15.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/OnOffResURI		get	post		

3167 **8.16 Oven Cycle Phase Mapping**

3168 **8.16.1 Introduction**

3169 This API defines the mapping between an instance of an AllJoyn OvenCyclePhase interface and
 3170 the OCF OperationalState Resource. OvenCyclePhase cyclephase Property pre-defines values
 3171 0x00-0x7F, 0x80-0xFF is for vendor specific values. The mapping defined herein covers only Spec
 3172 defined values. Any vendor defined value shall be represented in OCF using the x.<organization>
 3173 syntax for a vendor defined Property. The AllJoyn interface also supports a Method,
 3174 GetVendorPhasesDescription; this is handled in OCF using an instance of oic.r.actuator within an
 3175 oic.r.action collection. Please see Section 8 of the Mapping Specification for specifics.

3176 **8.16.2 Example URI**

3177 /OvenCyclePhaseResURI

3178 **8.16.3 Resource Type**

3179 The resource type (rt) is defined as: oic.r.operationalstate.

3180 **8.16.4 RAML Definition**

```

3181 #%RAML 0.8
3182 title: OvenCyclePhaseInterfaceMapping
3183 version: OCFv1.0.0-20170317
3184 traits:
3185   - interface-sensor :
3186     queryParameters:
3187       if:
3188         enum: ["oic.if.s", "oic.if.baseline"]
3189
3190 /OvenCyclePhaseResURI:
3191   description: |
3192     This API defines the mapping between an instance of an AllJoyn OvenCyclePhase interface and the
3193     OCF OperationalState Resource.
3194     OvenCyclePhase cyclephase Property pre-defines values 0x00-0x7F, 0x80-0xFF is for vendor
3195     specific values
3196     The mapping defined herein covers only Spec defined values.
3197     Any vendor defined value shall be represented in OCF using the x.<organization> syntax for a
3198     vendor defined Property.
3199     The AllJoyn interface also supports a Method, GetVendorPhasesDescription; this is handled in
3200     OCF using an instance of oic.r.actuator within an oic.r.action collection.
3201     Please see Section 8 of the Mapping Specification for specifics.
3202
3203   is : ['interface-sensor']
3204   get:
3205     responses :
3206       200:
3207         body:
3208           application/json:
3209             schema: /
  
```

```

3210     {
3211         "id":
3212         "http://openinterconnect.org/asamapping/schemas/asa.operation.ovencyclephase.json#",
3213         "$schema": "http://json-schema.org/draft-04/schema#",
3214         "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
3215 reserved.",
3216         "title": "Oven Cycle Phase",
3217         "definitions": {
3218             "asa.operation.ovencyclephase": {
3219                 "type": "object",
3220                 "properties": {
3221                     "cyclephase": {
3222                         "type": "integer",
3223                         "description": "Current phase of the operational cycle",
3224                         "x-ocf-conversion": {
3225                             "x-ocf-alias": "oic.r.operationalstate",
3226                             "x-to-ocf": [
3227                                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
3228                                 "currentmachinestate = phasearray[cyclephase]"
3229                             ],
3230                             "x-from-ocf": [
3231                                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
3232                                 "cyclephase = indexof statearray[currentmachinestate[0]]"
3233                             ]
3234                         }
3235                     },
3236                     "supportedcyclephases": {
3237                         "type": "array",
3238                         "items": {
3239                             "type": "integer"
3240                         },
3241                         "description": "Array of cycle phases supported by the Appliance.",
3242                         "x-ocf-conversion": {
3243                             "x-ocf-alias": "oic.r.operationalstate",
3244                             "x-to-ocf": [
3245                                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
3246                                 "for x=0, x < sizeof(supportedcyclephases): machinestates[x] =
3247 phasearray[supportedcyclephases[x]]"
3248                             ],
3249                             "x-from-ocf": [
3250                                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
3251                                 "for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof
3252 phasearray[machinestates[x]]"
3253                             ]
3254                         }
3255                     },
3256                     "getvendorphasesdescription": {
3257                         "x-ocf-type": "method",
3258                         "description": "Get cycle phases description",
3259                         "x-ocf-conversion": {
3260                             "x-ocf-alias": "oic.r.action"
3261                         }
3262                     }
3263                 }
3264             }
3265         },
3266         "type": "object",
3267         "allOf": [
3268             {"$ref": "#/definitions/asa.operation.ovencyclephase"}
3269         ],
3270         "required": [ "cyclephase","supportedcyclephases" ]
3271     }
3272
3273     example: /
3274     {
3275         "rt":      ["oic.r.operationalstate"]
3276     }
3277

```

3278

8.16.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Desc ription
supportedcyclephases	oic.r.operationalstate	phasearray = [Unavailable,Preheating,Cooking,Cleaning]for x=0, x < sizeof(supportedcyclephases): machinestates[x] = phasearray[supportedcyclephases[x]]	phasearray = [Unavailable,Preheating,Cooking,Cleaning]for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof phasearray[machinestates[x]]	Array of cycle phases supported by the Appliance.
cyclephase	oic.r.operationalstate	phasearray = [Unavailable,Preheating,Cooking,Cleaning]currentmachinestate = phasearray[cyclephase]	phasearray = [Unavailable,Preheating,Cooking,Cleaning]cyclephase = indexof statearray[currentmachinestate[0]]	Current phase of the operational cycle
getvendorphasedescription	oic.r.action			Get cycle phases description

3279

8.16.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/OvenCyclePhaseResURI		get			

3280

Annex A Swagger2.0 (Informative)

3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340

A.1 Audio Volume Mapping

A.1.1 Introduction

This API defines the mapping between an instance of an OCF Audio Controls and the AllJoyn Audio Volume interface.

A.1.2 Example URI

/AudioVolumeResURI

A.1.3 Resource Type

The resource type (rt) is defined as: ['oic.r.audio'].

A.1.4 Swagger2.0 Definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Audio Volume Mapping",
    "version": "OCFv1.0.0-20170317",
    "license": {
      "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n      1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \\"AS IS\\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n\n      IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n\n      HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/AudioVolumeResURI" : {
      "get": {
        "description": "This API defines the mapping between an instance of an OCF Audio
Controls\nand the AllJoyn Audio Volume interface.\n",
        "parameters": [
          { "$ref": "#/parameters/interface-all" }
        ],
        "responses": {
          "200": {
            "description": "",
            "x-example":
              {
                "rt": ["oic.r.audio"]
              }
          },
          "schema": { "$ref": "#/definitions/RetrieveSchema" }
        }
      }
    }
  }
},
```

```

3341     "post": {
3342       "description": "",
3343       "parameters": [
3344         { "$ref": "#/parameters/interface-all"},
3345         {
3346           "name": "body",
3347           "in": "body",
3348           "required": true,
3349           "schema": { "$ref": "#/definitions/UpdateSchema" }
3350         }
3351       ],
3352       "responses": {
3353         "200": {
3354           "description": "",
3355           "schema": { "$ref": "#/definitions/UpdateSchema" }
3356         }
3357       }
3358     }
3359   },
3360 },
3361 "parameters": {
3362   "interface-all": {
3363     "in": "query",
3364     "name": "if",
3365     "type": "string",
3366     "enum": ["oic.if.a", "oic.if.baseline"]
3367   }
3368 },
3369 "definitions": {
3370   "RetrieveSchema": {
3371     {
3372       "properties": {
3373         "maxvolume": {
3374           "type": "integer",
3375           "x-ocf-conversion": {
3376             "x-from-ocf": [
3377               "maxvolume = range[1]",
3378               "otherwise: maxvalue = 100"
3379             ],
3380             "x-ocf-alias": "oic.r.audio",
3381             "x-to-ocf": [
3382               "range[0] = 0",
3383               "range[1] = maxvolume"
3384             ]
3385           }
3386         },
3387         "mute": {
3388           "type": "boolean",
3389           "x-ocf-conversion": {
3390             "x-from-ocf": [
3391               "mute = ocf.mute"
3392             ],
3393             "x-ocf-alias": "oic.r.audio",
3394             "x-to-ocf": [
3395               "ocf.mute = mute"
3396             ]
3397           }
3398         },
3399         "volume": {
3400           "description": "Speaker volume index",
3401           "type": "integer",
3402           "x-ocf-conversion": {
3403             "x-from-ocf": [
3404               "volume = ocf.volume"
3405             ],
3406             "x-ocf-alias": "oic.r.audio",
3407             "x-to-ocf": [
3408               "ocf.volume = volume"
3409             ]
3410           }
3411         }
3412       }
3413     }
3414   }
3415 }

```



```

3412     },
3413     "required": [
3414         "volume",
3415         "maxvolume",
3416         "mute"
3417     ],
3418     "type": "object"
3419 }
3420
3421 ,
3422 "UpdateSchema" :
3423 {
3424     "properties": {
3425         "maxvolume": {
3426             "type": "integer",
3427             "x-ocf-conversion": {
3428                 "x-from-ocf": [
3429                     "maxvolume = range[1]",
3430                     "otherwise: maxvalue = 100"
3431                 ],
3432                 "x-ocf-alias": "oic.r.audio",
3433                 "x-to-ocf": [
3434                     "range[0] = 0",
3435                     "range[1] = maxvolume"
3436                 ]
3437             }
3438         },
3439         "mute": {
3440             "type": "boolean",
3441             "x-ocf-conversion": {
3442                 "x-from-ocf": [
3443                     "mute = ocf.mute"
3444                 ],
3445                 "x-ocf-alias": "oic.r.audio",
3446                 "x-to-ocf": [
3447                     "ocf.mute = mute"
3448                 ]
3449             }
3450         },
3451         "volume": {
3452             "description": "Speaker volume index",
3453             "type": "integer",
3454             "x-ocf-conversion": {
3455                 "x-from-ocf": [
3456                     "volume = ocf.volume"
3457                 ],
3458                 "x-ocf-alias": "oic.r.audio",
3459                 "x-to-ocf": [
3460                     "ocf.volume = volume"
3461                 ]
3462             }
3463         }
3464     },
3465     "required": [
3466         "volume",
3467         "maxvolume",
3468         "mute"
3469     ],
3470     "type": "object"
3471 }
3472 }
3473 }
3474 }
3475

```

3476 A.1.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
------------------------------	--------------	--------	----------	-------------

volume	oic.r.audio	ocf.volume = volume	ocf.volume = volume	Speaker volume index
maxvolume	oic.r.audio	range[0] = Orange[1] = maxvolume	maxvolume = range[1] otherwise: maxvalue = 100	
mute	oic.r.audio	ocf.mute = mute	mute = ocf.mute	

3477 **A.1.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/AudioVolumeResURI		get	post		

3478 **A.2 Climate Control Mode Mapping**

3479 **A.2.1 Introduction**

3480 This API defines the mapping between an instance of an AllJoyn ClimateControlMode interface
3481 and the OCF equivalent Resource.
3482 ClimateControlMode has three Properties; these map as follows:
3483 mode, supportedmodes -> Mode Resource
3484 operationalstate -> OperationalState Resource
3485 This can be represented in OCF either as two distinct Resource instances or a single instance with
3486 two RTs (oic.r.mode, oic.r.operationalstate)
3487

3488 **A.2.2 Example URI**

3489 /ClimateControlModeResURI

3490 **A.2.3 Resource Type**

3491 The resource type (rt) is defined as: ['oic.r.mode', 'oic.operationalstate'].

3492 **A.2.4 Swagger2.0 Definition**

```

3493 {
3494   "swagger": "2.0",
3495   "info": {
3496     "title": "Climate Control Mode Mapping",
3497     "version": "OCFv1.0.0-20170317",
3498     "license": {
3499       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
3500       "x-description": "Redistribution and use in source and binary forms, with or without
3501 modification, are permitted provided that the following conditions are met:\n      1.
3502 Redistributions of source code must retain the above copyright notice, this list of conditions and
3503 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
3504 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
3505 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
3506 Connectivity Foundation, INC. \\"AS IS\\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
3507 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
3508 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
3509 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
3510 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
3511 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND
3512 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
3513 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
3514 OF SUCH DAMAGE.\n"
3515     }
3516   },
3517   "schemes": ["http"],
3518   "consumes": ["application/json"],
3519   "produces": ["application/json"],
3520   "paths": {
3521     "/ClimateControlModeResURI" : {
3522       "get": {
3523         "description": "This API defines the mapping between an instance of an AllJoyn
3524 ClimateControlMode interface and the OCF equivalent Resource.\nClimateControlMode has three
3525 Properties; these map as follows:\n mode, supportedmodes -> Mode Resource\n operationalstate ->

```

```

3526 OperationalState Resource\nThis can be represented in OCF either as two distinct Resource instances
3527 or a single instance with two RTs (oic.r.mode, oic.r.operationalstate)\n",
3528     "parameters": [
3529         {"$ref": "#/parameters/interface-actuator"}
3530     ],
3531     "responses": {
3532         "200": {
3533             "description": "",
3534             "x-example":
3535                 {
3536                     "rt": ["oic.r.mode", "oic.operationalstate"]
3537                 }
3538             ,
3539             "schema": { "$ref": "#/definitions/RetrieveSchema" }
3540         }
3541     }
3542 },
3543 "post": {
3544     "description": "",
3545     "parameters": [
3546         {"$ref": "#/parameters/interface-actuator"},
3547         {
3548             "name": "body",
3549             "in": "body",
3550             "required": true,
3551             "schema": { "$ref": "#/definitions/UpdateSchema" }
3552         }
3553     ],
3554     "responses": {
3555         "200": {
3556             "description": "",
3557             "schema": { "$ref": "#/definitions/UpdateSchema" }
3558         }
3559     }
3560 }
3561 },
3562 },
3563 "parameters": {
3564     "interface-actuator" : {
3565         "in" : "query",
3566         "name" : "if",
3567         "type" : "string",
3568         "enum" : ["oic.if.a", "oic.if.baseline"]
3569     }
3570 },
3571 "definitions": {
3572     "RetrieveSchema" :
3573     {
3574         "properties": {
3575             "mode": {
3576                 "description": "Current mode of device.",
3577                 "type": "integer",
3578                 "x-ocf-conversion": {
3579                     "x-from-ocf": [
3580                         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3581                         "mode = indexof modeArray[ocf.mode[0]]"
3582                     ],
3583                     "x-ocf-alias": "oic.r.mode",
3584                     "x-to-ocf": [
3585                         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3586                         "ocf.mode[0] = modearray[mode]"
3587                     ]
3588                 }
3589             },
3590             "operationalstate": {
3591                 "description": "Current status of device",
3592                 "type": "integer",
3593                 "x-ocf-conversion": {
3594                     "x-from-ocf": [
3595                         "statearray = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
3596                         "operationalstate = indexof statearray[currentmachinestate[0]]"

```

```

3597         ],
3598         "x-ocf-alias": "oic.r.operationalstate",
3599         "x-to-ocf": [
3600             "machinestates = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
3601             "currentmachinestate = machinestates[operationalstate]"
3602         ]
3603     },
3604 },
3605 "supportedmodes": {
3606     "description": "Array of supported modes",
3607     "items": {
3608         "type": "integer"
3609     },
3610     "type": "array",
3611     "x-ocf-conversion": {
3612         "x-from-ocf": [
3613             "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3614             "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
3615 modearray[ocf.supportedmodes[x]]"
3616         ],
3617         "x-ocf-alias": "oic.r.mode",
3618         "x-to-ocf": [
3619             "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3620             "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
3621 modearray[supportedmodes[x]]"
3622         ]
3623     }
3624 },
3625 },
3626 "required": [
3627     "mode",
3628     "supportedmodes",
3629     "operationalstate"
3630 ],
3631 "type": "object"
3632 }
3633
3634 /
3635 "UpdateSchema" :
3636 {
3637     "properties": {
3638         "mode": {
3639             "description": "Current mode of device.",
3640             "type": "integer",
3641             "x-ocf-conversion": {
3642                 "x-from-ocf": [
3643                     "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3644                     "mode = indexof modeArray[ocf.mode[0]]"
3645                 ],
3646                 "x-ocf-alias": "oic.r.mode",
3647                 "x-to-ocf": [
3648                     "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3649                     "ocf.mode[0] = modearray[mode]"
3650                 ]
3651             }
3652         },
3653         "operationalstate": {
3654             "description": "Current status of device",
3655             "type": "integer",
3656             "x-ocf-conversion": {
3657                 "x-from-ocf": [
3658                     "statearray = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
3659                     "operationalstate = indexof statearray[currentmachinestate[0]]"
3660                 ],
3661                 "x-ocf-alias": "oic.r.operationalstate",
3662                 "x-to-ocf": [
3663                     "machinestates = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
3664                     "currentmachinestate = machinestates[operationalstate]"
3665                 ]
3666             }
3667         }
3668     }
3669 }

```

```

3668     "supportedmodes": {
3669         "description": "Array of supported modes",
3670         "items": {
3671             "type": "integer"
3672         },
3673         "type": "array",
3674         "x-ocf-conversion": {
3675             "x-from-ocf": [
3676                 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3677                 "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
3678 modearray[ocf.supportedmodes[x]]"
3679             ],
3680             "x-ocf-alias": "oic.r.mode",
3681             "x-to-ocf": [
3682                 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3683                 "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
3684 modearray[supportedmodes[x]]"
3685             ]
3686         }
3687     },
3688     "required": [
3689         "mode",
3690         "supportedmodes",
3691         "operationalstate"
3692     ],
3693     "type": "object"
3694 }
3695 }
3696 }
3697 }
3698 }
3699
3700

```

A.2.5 Property Definition

['AllJ oyn'] Property name	OCF Resource	To OCF	From OCF	Description
supportedmodes	oic.r.mode	modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] = modearray[supportedmodes[x]]	modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof modearray[ocf.supportedmodes[x]]	Array of supported modes
operationalstate	oic.r.operationalstate	machinestates = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]currentmachinestate = machinestates[operationalstate]	statearray = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]operationalstate = indexof statearray[currentmachinestate[0]]	Current status of device
mode	oic.r.mode	modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]ocf.mode[0] = modearray[mode]	modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]mode = indexof modeArray[ocf.mode[0]]	Current mode of device.

A.2.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/ClimateControlModeResURI		get	post		

3702 A.3 Closed Status Mapping

3703 A.3.1 Introduction

3704 This API defines the mapping between an instance of an AllJoyn ClosedStatus Interface and
3705 the OCF Door Resource.
3706

3707 A.3.2 Example URI

3708 /ClosedStatusResURI

3709 A.3.3 Resource Type

3710 The resource type (rt) is defined as: ['oic.r.door'].

3711 A.3.4 Swagger2.0 Definition

```
3712 {  
3713   "swagger": "2.0",  
3714   "info": {  
3715     "title": "Closed Status Mapping",  
3716     "version": "OCFv1.0.0-20170317",  
3717     "license": {  
3718       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",  
3719       "x-description": "Redistribution and use in source and binary forms, with or without  
3720 modification, are permitted provided that the following conditions are met:\n      1.  
3721 Redistributions of source code must retain the above copyright notice, this list of conditions and  
3722 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above  
3723 copyright notice, this list of conditions and the following disclaimer in the documentation and/or  
3724 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open  
3725 Connectivity Foundation, INC. \AS IS\ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT  
3726 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR  
3727 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n\n      IN NO EVENT SHALL THE Open Connectivity  
3728 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,  
3729 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
3730 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n\n      HOWEVER CAUSED AND  
3731 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR  
3732 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY  
3733 OF SUCH DAMAGE.\n\n    }  
3734   },  
3735   "schemes": ["http"],  
3736   "consumes": ["application/json"],  
3737   "produces": ["application/json"],  
3738   "paths": {  
3739     "/ClosedStatusResURI" : {  
3740       "get": {  
3741         "description": "This API defines the mapping between an instance of an AllJoyn ClosedStatus  
3742 Interface and\nthe OCF Door Resource.\n",  
3743         "parameters": [  
3744           { "$ref": "#/parameters/interface-all" }  
3745         ],  
3746         "responses": {  
3747           "200": {  
3748             "description" : "",  
3749             "x-example":  
3750               {  
3751                 "rt": ["oic.r.door"]  
3752               }  
3753             },  
3754             "schema": { "$ref": "#/definitions/RetrieveSchema" }  
3755           }  
3756         }  
3757       }  
3758     }  
3759   },  
3760   "parameters": {  
3761     "interface-all" : {  
3762       "in" : "query",  
3763       "name" : "if",  
3764
```

```

3765     "type" : "string",
3766     "enum" : ["oic.if.s", "oic.if.baseline"]
3767   },
3768 },
3769 "definitions": {
3770   "RetrieveSchema" :
3771     {
3772     "properties": {
3773       "isclosed": {
3774         "description": "Open/Closed status Indicator",
3775         "type": "boolean",
3776         "x-ocf-conversion": {
3777           "x-from-ocf": [
3778             "isClosed = (openState == Closed)"
3779           ],
3780           "x-ocf-alias": "oic.r.door",
3781           "x-to-ocf": [
3782             "if isClosed ocf.openState = Closed.",
3783             "if !isClosed ocf.openState = Open."
3784           ]
3785         }
3786       }
3787     },
3788     "required": [
3789       "isclosed"
3790     ],
3791     "type": "object"
3792   }
3793 }
3794 }
3795 }
3796

```

3797 A.3.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
isclosed	oic.r.door	if isClosed ocf.openState = Closed.if !isClosed ocf.openState = Open.	isClosed = (openState == Closed)	Open/Closed status Indicator

3798 A.3.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/ClosedStatusResURI		get			

3799 A.4 Air Quality Mapping

3800 A.4.1 Introduction

3801 This API defines the mapping between the AllJoyn AirQuality interface and the OCF AirQuality
3802 Resource.
3803 If more than one instance of the AirQuality interface is exposed then each instance maps to an
3804 instance of the OCF AirQuality Resource.
3805 The mapping defined in the schema describes the population of the OCF AirQuality Resource.
3806 Even if there is only a single instance of an OCF AirQuality Resource this shall be included in an
3807 instance of an OCF AirQualityCollection.
3808 The number of links in the collection equates to the number of instances of the AllJoyn
3809 CurrentAirQuality interface that are exposed.
3810 When mapping from OCF the valueType of the Resource shall be introspected, this API is invoked
3811 only if this is set to 'Measured'
3812

3813 **A.4.2 Example URI**

3814 /CurrentAirQualityResURI

3815 **A.4.3 Resource Type**

3816 The resource type (rt) is defined as: ['oic.r.airqualitycollection'].

3817 **A.4.4 Swagger2.0 Definition**

```
3818 {
3819   "swagger": "2.0",
3820   "info": {
3821     "title": "Air Quality Mapping",
3822     "version": "OCFv1.0.0-20170317",
3823     "license": {
3824       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
3825       "x-description": "Redistribution and use in source and binary forms, with or without
3826 modification, are permitted provided that the following conditions are met:\n      1.
3827 Redistributions of source code must retain the above copyright notice, this list of conditions and
3828 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
3829 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
3830 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
3831 Connectivity Foundation, INC. \\"AS IS\\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
3832 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
3833 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
3834 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
3835 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
3836 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND
3837 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
3838 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
3839 OF SUCH DAMAGE.\n"
3840   }
3841 },
3842 "schemes": ["http"],
3843 "consumes": ["application/json"],
3844 "produces": ["application/json"],
3845 "paths": {
3846   "/CurrentAirQualityResURI" : {
3847     "get": {
3848       "description": "This API defines the mapping between the AllJoyn AirQuality interface and
3849 the OCF AirQuality Resource.\nIf more than one instance of the AirQuality interface is exposed then
3850 each instance maps to an instance of the OCF AirQuality Resource.\nThe mapping defined in the
3851 schema describes the population of the OCF AirQuality Resource.\nEven if there is only a single
3852 instance of an OCF AirQuality Resource this shall be included in an instance of an OCF
3853 AirQualityCollection.\nThe number of links in the collection equates to the number of instances of
3854 the AllJoyn CurrentAirQuality interface that are exposed.\nWhen mapping from OCF the valueType of
3855 the Resource shall be introspected, this API is invoked only if this is set to 'Measured'\n",
3856     "parameters": [
3857       { "$ref": "#/parameters/interface-sensor" }
3858     ],
3859     "responses": {
3860       "200": {
3861         "description": "",
3862         "x-example":
3863           {
3864             "rt": ["oic.r.airqualitycollection"]
3865           }
3866       },
3867       "schema": { "$ref": "#/definitions/RetrieveSchema" }
3868     }
3869   }
3870 }
3871 },
3872 },
3873 "parameters": {
3874   "interface-sensor" : {
3875     "in" : "query",
3876     "name" : "if",
3877     "type" : "string",
3878     "enum" : ["oic.if.s", "oic.if.baseline"]
3879   }
3880 }
```



```

3880 },
3881 "definitions": {
3882   "RetrieveSchema" :
3883     {
3884       "properties": {
3885         "contaminanttype": {
3886           "description": "The contaminant type",
3887           "type": "integer",
3888           "x-ocf-conversion": {
3889             "x-from-ocf": [
3890               "contaminanttype = indexof contaminanttypearray[ocf.contaminanttype]"
3891             ],
3892             "x-ocf-alias": "oic.r.airquality",
3893             "x-to-ocf": [
3894               "valuetype = Measured",
3895               "contaminanttypearray = [CH2O,CO2,CO,PM2_5,PM10,VOC]",
3896               "ocf.contaminanttype = contaminanttypearray[contaminanttype]"
3897             ]
3898           }
3899         },
3900         "currentvalue": {
3901           "type": "number",
3902           "x-ocf-conversion": {
3903             "x-from-ocf": [
3904               "currentvalue = contaminantvalue"
3905             ],
3906             "x-ocf-alias": "oic.r.airquality",
3907             "x-to-ocf": [
3908               "contaminantvalue = currentvalue"
3909             ]
3910           }
3911         },
3912         "maxvalue": {
3913           "type": "number",
3914           "x-ocf-conversion": {
3915             "x-from-ocf": [
3916               "maxvalue = range[1]"
3917             ],
3918             "x-ocf-alias": "oic.r.airquality",
3919             "x-to-ocf": [
3920               "range[1] = maxvalue"
3921             ]
3922           }
3923         },
3924         "minvalue": {
3925           "type": "number",
3926           "x-ocf-conversion": {
3927             "x-from-ocf": [
3928               "minvalue = range[0]"
3929             ],
3930             "x-ocf-alias": "oic.r.airquality",
3931             "x-to-ocf": [
3932               "range[0] = minvalue"
3933             ]
3934           }
3935         },
3936         "precision": {
3937           "type": "number",
3938           "x-ocf-conversion": {
3939             "x-from-ocf": [
3940               "precision = ocf.precision"
3941             ],
3942             "x-ocf-alias": "oic.r.airquality",
3943             "x-to-ocf": [
3944               "ocf.precision = precision"
3945             ]
3946           }
3947         },
3948         "updatemintime": {
3949           "type": "integer",
3950           "x-ocf-conversion": {

```

```

3951         "x-from-ocf": [
3952             "updatemintime = ocf.minnotifyperiod"
3953         ],
3954         "x-ocf-alias": "oic.r.value.conditional",
3955         "x-to-ocf": [
3956             "ocf.minnotifyperiod = updatemintime"
3957         ]
3958     }
3959 },
3960 },
3961     "required": [
3962         "contaminanttype",
3963         "currentvalue",
3964         "minvalue",
3965         "maxvalue",
3966         "precision",
3967         "updatemintime"
3968     ],
3969     "type": "object"
3970 }
3971 }
3972 }
3973 }
3974 }

```

3975 A.4.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
currentvalue	oic.r.airquality	contaminantvalue = currentvalue	currentvalue = contaminantvalue	
updatemintime	oic.r.value.conditional	ocf.minnotifyperiod = updatemintime	updatemintime = ocf.minnotifyperiod	
maxvalue	oic.r.airquality	range[1] = maxvalue	maxvalue = range[1]	
precision	oic.r.airquality	ocf.precision = precision	precision = ocf.precision	
minvalue	oic.r.airquality	range[0] = minvalue	minvalue = range[0]	
contaminanttype	oic.r.airquality	valuetype = Measuredcontaminanttypearray = [CH2O,CO2,CO,PM2_5,PM10,VO]ocf.contaminanttype = contaminanttypearray[contaminanttype]	contaminanttype = indexof contaminanttypearray[ocf.contaminanttype]	The contaminant type

3976 A.4.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentAirQualityResURI		get			

3977 A.5 Air Quality Level Mapping

3978 A.5.1 Introduction

3979 This API defines the mapping between the AllJoyn AirQualityLevel interface and the OCF
3980 AirQuality Resource.
3981 If more than one instance of the AirQualityLevel interface is exposed then each instance maps to
3982 an instance of the OCF AirQuality Resource.
3983 The mapping defined in the schema describes the population of the OCF AirQuality Resource.

3984 Even if there is only a single instance of an OCF AirQuality Resource then this shall be included
3985 in an instance of an OCF AirQualityCollection.
3986 The number of links in the collection equates to the number of instances of the AllJoyn
3987 CurrentAirQuality interface that are exposed.
3988 When mapping from OCF the valueType of the Resource shall be introspected, this API is invoked
3989 only if this is set to 'Qualitative'
3990

3991 **A.5.2 Example URI**

3992 /CurrentAirQualityLevelResURI

3993 **A.5.3 Resource Type**

3994 The resource type (rt) is defined as: ['oic.r.airqualitycollection'].

3995 **A.5.4 Swagger2.0 Definition**

```
3996 {  
3997   "swagger": "2.0",  
3998   "info": {  
3999     "title": "Air Quality Level Mapping",  
4000     "version": "OCFv1.0.0-20170317",  
4001     "license": {  
4002       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",  
4003       "x-description": "Redistribution and use in source and binary forms, with or without  
4004 modification, are permitted provided that the following conditions are met:\n      1.  
4005 Redistributions of source code must retain the above copyright notice, this list of conditions and  
4006 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above  
4007 copyright notice, this list of conditions and the following disclaimer in the documentation and/or  
4008 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open  
4009 Connectivity Foundation, INC. \AS IS\ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT  
4010 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR  
4011 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n\n      IN NO EVENT SHALL THE Open Connectivity  
4012 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,  
4013 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
4014 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n\n      HOWEVER CAUSED AND  
4015 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR  
4016 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY  
4017 OF SUCH DAMAGE.\n\n    }  
4018   },  
4019   "schemes": ["http"],  
4020   "consumes": ["application/json"],  
4021   "produces": ["application/json"],  
4022   "paths": {  
4023     "/CurrentAirQualityLevelResURI" : {  
4024       "get": {  
4025         "description": "This API defines the mapping between the AllJoyn AirQualityLevel interface  
4026 and the OCF AirQuality Resource.\n\nIf more than one instance of the AirQualityLevel interface is  
4028 exposed then each instance maps to an instance of the OCF AirQuality Resource.\n\nThe mapping defined  
4029 in the schema describes the population of the OCF AirQuality Resource.\n\nEven if there is only a  
4030 single instance of an OCF AirQuality Resource then this shall be included in an instance of an OCF  
4031 AirQualityCollection.\n\nThe number of links in the collection equates to the number of instances of  
4032 the AllJoyn CurrentAirQuality interface that are exposed.\n\nWhen mapping from OCF the valueType of  
4033 the Resource shall be introspected, this API is invoked only if this is set to 'Qualitative'\n",  
4034         "parameters": [  
4035           {"$ref": "#/parameters/interface-sensor"}  
4036         ],  
4037         "responses": {  
4038           "200": {  
4039             "description": "",  
4040             "x-example":  
4041               {  
4042                 "rt": ["oic.r.airqualitycollection"]  
4043               }  
4044             ,  
4045             "schema": {"$ref": "#/definitions/RetrieveSchema"}  
4046           }  
4047         }  
4048       }  
4049     }  
4050   }  
4051 }
```

```

4048     }
4049   },
4050 },
4051 "parameters": {
4052   "interface-sensor" : {
4053     "in" : "query",
4054     "name" : "if",
4055     "type" : "string",
4056     "enum" : ["oic.if.s", "oic.if.baseline"]
4057   }
4058 },
4059 "definitions": {
4060   "RetrieveSchema" :
4061     {
4062     "properties": {
4063       "contaminanttype": {
4064         "description": "The contaminant type",
4065         "type": "integer",
4066         "x-ocf-conversion": {
4067           "x-from-ocf": [
4068             "if ocf.contaminanttype = CH2O, contaminanttype = 0",
4069             "if ocf.contaminanttype = CO2, contaminanttype = 1",
4070             "if ocf.contaminanttype = CO, contaminanttype = 2",
4071             "if ocf.contaminanttype = PM2_5, contaminanttype = 3",
4072             "if ocf.contaminanttype = PM10, contaminanttype = 4",
4073             "if ocf.contaminanttype = VOC, contaminanttype = 5",
4074             "if ocf.contaminanttype = Smoke, contaminanttype = 253",
4075             "if ocf.contaminanttype = Odor, contaminanttype = 254",
4076             "if ocf.contaminanttype = AirPollution, contaminanttype = 255"
4077           ],
4078           "x-ocf-alias": "oic.r.airquality",
4079           "x-to-ocf": [
4080             "valuetype = Qualitative",
4081             "if contaminanttype = 0, ocf.contaminanttype = CH2O",
4082             "if contaminanttype = 1, ocf.contaminanttype = CO2",
4083             "if contaminanttype = 2, ocf.contaminanttype = CO",
4084             "if contaminanttype = 3, ocf.contaminanttype = PM2_5",
4085             "if contaminanttype = 4, ocf.contaminanttype = PM10",
4086             "if contaminanttype = 5, ocf.contaminanttype = VOC",
4087             "if contaminanttype = 253, ocf.contaminanttype = Smoke",
4088             "if contaminanttype = 254, ocf.contaminanttype = Odor",
4089             "if contaminanttype = 255, ocf.contaminanttype = AirPollution"
4090           ]
4091         }
4092       },
4093       "currentlevel": {
4094         "type": "integer",
4095         "x-ocf-conversion": {
4096           "x-from-ocf": [
4097             "currentlevel = contaminantvalue"
4098           ],
4099           "x-ocf-alias": "oic.r.airquality",
4100           "x-to-ocf": [
4101             "contaminantvalue = currentlevel"
4102           ]
4103         }
4104       },
4105       "maxlevel": {
4106         "type": "integer",
4107         "x-ocf-conversion": {
4108           "x-from-ocf": [
4109             "maxvalue = range[1]"
4110           ],
4111           "x-ocf-alias": "oic.r.airquality",
4112           "x-to-ocf": [
4113             "range[0] = 0",
4114             "range[1] = maxvalue"
4115           ]
4116         }
4117       }
4118     }

```

```

4119     "required": [
4120         "contaminanttype",
4121         "currentlevel",
4122         "maxlevel"
4123     ],
4124     "type": "object"
4125 }
4126 }
4127 }
4128 }
4129 }

```

4130 A.5.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
currentlevel	oic.r.airquality	contaminantvalue = currentlevel	currentlevel = contaminantvalue	
maxlevel	oic.r.airquality	range[0] = 0range[1] = maxvalue	maxvalue = range[1]	
contaminanttype	oic.r.airquality	valuetype = Qualitativeif contaminanttype = 0, ocf.contaminanttype = CH2Oif contaminanttype = 1, ocf.contaminanttype = CO2if contaminanttype = 2, ocf.contaminanttype = COif contaminanttype = 3, ocf.contaminanttype = PM2_5if contaminanttype = 4, ocf.contaminanttype = PM10if contaminanttype = 5, ocf.contaminanttype = VOCif contaminanttype = 253, ocf.contaminanttype = Smokeif contaminanttype = 254, ocf.contaminanttype = Odorif contaminanttype = 255, ocf.contaminanttype = AirPollution	if ocf.contaminanttype = CH2O, contaminanttype = 0if ocf.contaminanttype = CO2, contaminanttype = 1if ocf.contaminanttype = CO, contaminanttype = 2if ocf.contaminanttype = PM2_5, contaminanttype = 3if ocf.contaminanttype = PM10, contaminanttype = 4if ocf.contaminanttype = VOC, contaminanttype = 5if ocf.contaminanttype = Smoke, contaminanttype = 253if ocf.contaminanttype = Odor, contaminanttype = 254if ocf.contaminanttype = AirPollution, contaminanttype = 255	The contaminant type

4131 **A.5.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/CurrentAirQualityLevelResURI		get			

4132 **A.6 Current Humidity Mapping**

4133 **A.6.1 Introduction**

4134 This API defines the mapping between an instance of an OCF Humidity which exposes only a
 4135 sensor interface
 4136 and the AllJoyn Current Humidity interface.
 4137 A RETRIEVE on a Temperature Sensor maps to an action on an instance of an
 4138 Environment.CurrentTemperature Interface.
 4139

4140 **A.6.2 Example URI**

4141 /CurrentHumidityResURI

4142 **A.6.3 Resource Type**

4143 The resource type (rt) is defined as: ['oic.r.humidity'].

4144 **A.6.4 Swagger2.0 Definition**

```

4145 {
4146   "swagger": "2.0",
4147   "info": {
4148     "title": "Current Humidity Mapping",
4149     "version": "OCFv1.0.0-20170317",
4150     "license": {
4151       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
4152       "x-description": "Redistribution and use in source and binary forms, with or without
4153 modification, are permitted provided that the following conditions are met:\n      1.
4154 Redistributions of source code must retain the above copyright notice, this list of conditions and
4155 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
4156 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
4157 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
4158 Connectivity Foundation, INC. \AS IS\ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
4159 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
4160 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
4161 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
4162 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
4163 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND
4164 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
4165 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
4166 OF SUCH DAMAGE.\n\n"
4167     }
4168   },
4169   "schemes": ["http"],
4170   "consumes": ["application/json"],
4171   "produces": ["application/json"],
4172   "paths": {
4173     "/CurrentHumidityResURI" : {
4174       "get": {
4175         "description": "This API defines the mapping between an instance of an OCF Humidity which
4176 exposes only a sensor interface\nand the AllJoyn Current Humidity interface.\nA RETRIEVE on a
4177 Temperature Sensor maps to an action on an instance of an Environment.CurrentTemperature
4178 Interface.\n",
4179         "parameters": [
4180           {"$ref": "#/parameters/interface-sensor"}
4181         ],
4182         "responses": {
4183           "200": {
4184             "description": "",
4185             "x-example":
4186               {
4187                 "rt": ["oic.r.humidity"]
4188               }
4189           }
4190         }
4191       }
4192     }
4193   }
4194 }
  
```

```

4189         'schema': { "$ref": "#/definitions/RetrieveSchema" }
4190     }
4191 }
4192 }
4193 }
4194 }
4195 },
4196 "parameters": {
4197     "interface-sensor" : {
4198         "in" : "query",
4199         "name" : "if",
4200         "type" : "string",
4201         "enum" : ["oic.if.s", "oic.if.baseline"]
4202     }
4203 },
4204 "definitions": {
4205     "RetrieveSchema" :
4206     {
4207         "properties": {
4208             "currentvalue": {
4209                 "description": "Measured value",
4210                 "type": "number",
4211                 "x-ocf-conversion": {
4212                     "x-from-ocf": [
4213                         "currentvalue = humidity"
4214                     ],
4215                     "x-ocf-alias": "oic.r.humidity",
4216                     "x-to-ocf": [
4217                         "humidity = currentValue"
4218                     ]
4219                 }
4220             },
4221             "maxvalue": {
4222                 "description": "Max measured value for humidty",
4223                 "type": "number",
4224                 "x-ocf-conversion": {
4225                     "x-from-ocf": [
4226                         "maxvalue = range[1]"
4227                     ],
4228                     "x-ocf-alias": "oic.r.humidity",
4229                     "x-to-ocf": [
4230                         "range[0] = 0",
4231                         "range[1] = maxvalue"
4232                     ]
4233                 }
4234             }
4235         },
4236         "required": [
4237             "currentvalue",
4238             "maxvalue"
4239         ],
4240         "type": "object"
4241     }
4242 }
4243 }
4244 }
4245

```

4246 A.6.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
currentvalue	oic.r.humidity	humidity currentValue	= currentvalue = humidity	Measured value
maxvalue	oic.r.humidity	range[0] Orange[1] maxvalue	= =	Max measured value for humidty

4247 **A.6.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/CurrentHumidityResURI		get			

4248 **A.7 Current Temperature Mapping**

4249 **A.7.1 Introduction**

4250 This API defines the mapping between an instance of an OCF Temperature which exposes only a
 4251 sensor interface
 4252 and the AllJoyn Current Temperature interface.
 4253 A RETRIEVE on a Temperature Sensor maps to an action on an instance of an
 4254 Environment.CurrentTemperature Interface.
 4255

4256 **A.7.2 Example URI**

4257 /CurrentTemperatureResURI

4258 **A.7.3 Resource Type**

4259 The resource type (rt) is defined as: ['oic.r.temperature'].

4260 **A.7.4 Swagger2.0 Definition**

```

4261 {
4262   "swagger": "2.0",
4263   "info": {
4264     "title": "Current Temperature Mapping",
4265     "version": "OCFv1.0.0-20170317",
4266     "license": {
4267       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
4268       "x-description": "Redistribution and use in source and binary forms, with or without
4269 modification, are permitted provided that the following conditions are met:\n      1.
4270 Redistributions of source code must retain the above copyright notice, this list of conditions and
4271 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
4272 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
4273 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
4274 Connectivity Foundation, INC. \AS IS\ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
4275 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
4276 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
4277 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
4278 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
4279 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND
4280 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
4281 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
4282 OF SUCH DAMAGE.\n"
4283     }
4284   },
4285   "schemes": ["http"],
4286   "consumes": ["application/json"],
4287   "produces": ["application/json"],
4288   "paths": {
4289     "/CurrentTemperatureResURI" : {
4290       "get": {
4291         "description": "This API defines the mapping between an instance of an OCF Temperature
4292 which exposes only a sensor interface\nand the AllJoyn Current Temperature interface.\nA RETRIEVE
4293 on a Temperature Sensor maps to an action on an instance of an Environment.CurrentTemperature
4294 Interface.\n",
4295         "parameters": [
4296           { "$ref": "#/parameters/interface-sensor" }
4297         ],
4298         "responses": {
4299           "200": {
4300             "description": "",
4301             "x-example":
4302               {
4303                 "rt": ["oic.r.temperature"]
4304               }
4305           }
4306         }
4307       }
4308     }
4309   }
4310 }
  
```



```

4305         'schema': { "$ref": "#/definitions/RetrieveSchema" }
4306     }
4307 }
4308 }
4309 }
4310 }
4311 },
4312 "parameters": {
4313     "interface-sensor" : {
4314         "in" : "query",
4315         "name" : "if",
4316         "type" : "string",
4317         "enum" : ["oic.if.s", "oic.if.baseline"]
4318     }
4319 },
4320 "definitions": {
4321     "RetrieveSchema" :
4322     {
4323         "properties": {
4324             "currentvalue": {
4325                 "description": "Measured value",
4326                 "type": "number",
4327                 "x-ocf-conversion": {
4328                     "x-from-ocf": {
4329                         "oneOf": [
4330                             {
4331                                 "properties": {
4332                                     "enum": [
4333                                         "C"
4334                                     ],
4335                                     "units": "string"
4336                                 },
4337                                 "x-from-ocf": [
4338                                     "currentvalue = temperature"
4339                                 ]
4340                             },
4341                             {
4342                                 "properties": {
4343                                     "enum": [
4344                                         "F"
4345                                     ],
4346                                     "units": "string"
4347                                 },
4348                                 "x-from-ocf": [
4349                                     "currentvalue = (temperature-32)*5/9"
4350                                 ]
4351                             },
4352                             {
4353                                 "properties": {
4354                                     "enum": [
4355                                         "K"
4356                                     ],
4357                                     "units": "string"
4358                                 },
4359                                 "x-from-ocf": [
4360                                     "currentvalue = temperature-273.15"
4361                                 ]
4362                             }
4363                         ]
4364                     },
4365                     "x-ocf-alias": "oic.r.temperature",
4366                     "x-to-ocf": [
4367                         "temperature = currentValue",
4368                         "units = C"
4369                     ]
4370                 }
4371             },
4372             "precision": {
4373                 "type": "number",
4374                 "x-ocf-conversion": {
4375                     "x-from-ocf": [

```

```

4376         "precision = ocf.precision"
4377     ],
4378     "x-ocf-alias": "oic.r.temperature",
4379     "x-to-ocf": [
4380         "ocf.precision = precision"
4381     ]
4382 }
4383 },
4384 "updatemintime": {
4385     "type": "integer",
4386     "x-ocf-conversion": {
4387         "x-from-ocf": [
4388             "updatemintime = ocf.minnotifyperiod"
4389         ],
4390         "x-ocf-alias": "oic.r.value.conditional",
4391         "x-to-ocf": [
4392             "ocf.minnotifyperiod = updatemintime"
4393         ]
4394     }
4395 }
4396 },
4397 "required": [
4398     "currentvalue",
4399     "precision",
4400     "updatemintime"
4401 ],
4402 "type": "object"
4403 }
4404 }
4405 }
4406 }
4407 }

```

4408 **A.7.5 Property Definition**

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
currentvalue	oic.r.temperature	temperature = currentValueunits = C	oneOf	Measured value
updatemintime	oic.r.value.conditiona l	ocf.minnotifyperio d = updatemintime	updatemintime = ocf.minnotifyperio d	
precision	oic.r.temperature	ocf.precision = precision	precision = ocf.precision	

4409 **A.7.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/CurrentTemperatureResURI		get			

4410 **A.8 Cycle Control Mapping**

4411 **A.8.1 Introduction**

4412 This API defines the mapping between an instance of an AllJoyn CycleControl interface and the
4413 OCF OperationalState Resource.
4414 The AllJoyn interface also supports a Method, ExecuteOperationalCommand; this is handled in
4415 OCF using an instance of oic.r.actuator within an oic.r.action collection.
4416 Please see Section 8 of the Mapping Specification for specifics.
4417

4418 **A.8.2 Example URI**

4419 /CycleControlResURI

4420 A.8.3 Resource Type

4421 The resource type (rt) is defined as: ['oic.r.operationalstate'].

4422 A.8.4 Swagger2.0 Definition

```
4423 {
4424   "swagger": "2.0",
4425   "info": {
4426     "title": "Cycle Control Mapping",
4427     "version": "OCFv1.0.0-20170317",
4428     "license": {
4429       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
4430       "x-description": "Redistribution and use in source and binary forms, with or without
4431 modification, are permitted provided that the following conditions are met:\n      1.
4432 Redistributions of source code must retain the above copyright notice, this list of conditions and
4433 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
4434 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
4435 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
4436 Connectivity Foundation, INC. \ "AS IS\ " AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
4437 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
4438 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
4439 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
4440 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
4441 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND
4442 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
4443 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
4444 OF SUCH DAMAGE.\n"
4445     }
4446   },
4447   "schemes": ["http"],
4448   "consumes": ["application/json"],
4449   "produces": ["application/json"],
4450   "paths": {
4451     "/CycleControlResURI" : {
4452       "get": {
4453         "description": "This API defines the mapping between an instance of an AllJoyn CycleControl
4454 interface and the OCF OperationalState Resource.\nThe AllJoyn interface also supports a Method,
4455 ExecuteOperationalCommand; this is handled in OCF using an instance of oic.r.actuator within an
4456 oic.r.action collection.\nPlease see Section 8 of the Mapping Specification for specifics.\n",
4457         "parameters": [
4458           { "$ref": "#/parameters/interface-sensor" }
4459         ],
4460         "responses": {
4461           "200": {
4462             "description": "",
4463             "x-example": {
4464               "rt": ["oic.r.operationalstate"]
4465             }
4466           },
4467           "schema": { "$ref": "#/definitions/RetrieveSchema" }
4468         }
4469       }
4470     }
4471   }
4472 },
4473 "parameters": {
4474   "interface-sensor" : {
4475     "in" : "query",
4476     "name" : "if",
4477     "type" : "string",
4478     "enum" : ["oic.if.s", "oic.if.baseline"]
4479   }
4480 },
4481 "definitions": {
4482   "RetrieveSchema" : {
4483     "properties": {
4484       "cyclephase": {
4485         "description": "Current phase of the operational cycle",
4486         "type": "integer",
4487       }
4488     }
4489   }
4490 }
```

```

4489     "x-ocf-conversion": {
4490         "x-from-ocf": [
4491             "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
4492             "cyclephase = indexof statearray[currentmachinestate[0]]"
4493         ],
4494         "x-ocf-alias": "oic.r.operationalstate",
4495         "x-to-ocf": [
4496             "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
4497             "currentmachinestate = phasearray[cyclephase]"
4498         ]
4499     },
4500 },
4501 "getvendorphasesdescription": {
4502     "description": "Get cycle phases description",
4503     "x-ocf-conversion": {
4504         "x-ocf-alias": "oic.r.action"
4505     },
4506     "x-ocf-type": "method"
4507 },
4508 "supportedcyclephases": {
4509     "description": "Array of cycle phases supported by the Appliance.",
4510     "items": {
4511         "type": "integer"
4512     },
4513     "type": "array",
4514     "x-ocf-conversion": {
4515         "x-from-ocf": [
4516             "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
4517             "for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof
4518 phasearray[machinestates[x]]"
4519         ],
4520         "x-ocf-alias": "oic.r.operationalstate",
4521         "x-to-ocf": [
4522             "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
4523             "for x=0, x < sizeof(supportedcyclephases): machinestates[x] =
4524 phasearray[supportedcyclephases[x]]"
4525         ]
4526     }
4527 },
4528 },
4529 "required": [
4530     "cyclephase",
4531     "supportedcyclephases"
4532 ],
4533 "type": "object"
4534 }
4535 }
4536 }
4537 }
4538

```

4539 A.8.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Desc ription
supportedcy clephases	oic.r.oper ationalsta te	phasearray = [Unavailable,Preheating,Cooki ng,Cleaning]for x=0, x < sizeof(supportedcyclephases): machinestates[x] = phasearray[supportedcycleph ases[x]]	phasearray = [Unavailable,Preheating, Cooking,Cleaning]for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof phasearray[machinestate s[x]]	Array of cycle phas es supp orted by the Appli ance.

cyclephase	oic.r.operationalstate	phasearray = [Unavailable,Preheating,Cooking,Cleaning]currentmachinestate = phasearray[cyclephase]	phasearray = [Unavailable,Preheating,Cooking,Cleaning]cyclephase = indexofstatearray[currentmachinestate[0]]	Current phase of the operational cycle
getvendorphasedescription	oic.r.action			Get cycle phases description

4540 **A.8.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/CycleControlResURI		get			

4541 **A.9 Fan Speed Level Mapping**

4542 **A.9.1 Introduction**

4543 This API defines the mapping between an instance of an AllJoyn FanSpeedLevel interface and an
 4544 OCF AirFlow Resource.
 4545 Note that the setting of the FanSpeedLevel to '0x00' (off) is handled via the 'OffControl' interface
 4546 rather than writing directly to this interface.
 4547 In such a case an instance of Binary Switch shall be exposed on the OCF side; this can be modeled
 4548 as AirFlowControl which is then a collection of Binary Switch and AirFlow.
 4549

4550 **A.9.2 Example URI**

4551 /FanSpeedLevelResURI

4552 **A.9.3 Resource Type**

4553 The resource type (rt) is defined as: ['oic.r.airflow'].

4554 **A.9.4 Swagger2.0 Definition**

```

4555 {
4556   "swagger": "2.0",
4557   "info": {
4558     "title": "Fan Speed Level Mapping",
4559     "version": "OCFv1.0.0-20170317",
4560     "license": {
4561       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
4562       "x-description": "Redistribution and use in source and binary forms, with or without
4563 modification, are permitted provided that the following conditions are met:\n      1.
4564 Redistributions of source code must retain the above copyright notice, this list of conditions and
4565 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
4566 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
4567 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
4568 Connectivity Foundation, INC. \AS IS\ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
4569 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
4570 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
4571 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
4572 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
4573 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND
4574 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
4575 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
4576 OF SUCH DAMAGE.\n"
4577   }
4578 },
  
```

```

4579     "schemes": ["http"],
4580     "consumes": ["application/json"],
4581     "produces": ["application/json"],
4582     "paths": {
4583         "/FanSpeedLevelResURI" : {
4584             "get": {
4585                 "description": "This API defines the mapping between an instance of an AllJoyn
4586 FanSpeedLevel interface and an OCF AirFlow Resource.\nNote that the setting of the FanSpeedLevel to
4587 '0x00' (off) is handled via the 'OffControl' interface rather than writing directly to this
4588 interface.\nIn such a case an instance of Binary Switch shall be exposed on the OCF side; this can
4589 be modeled as AirFlowControl which is then a collection of Binary Switch and AirFlow.\n",
4590                 "parameters": [
4591                     {"$ref": "#/parameters/interface-actuator"}
4592                 ],
4593                 "responses": {
4594                     "200": {
4595                         "description": "",
4596                         "x-example":
4597                             {
4598                                 "rt": ["oic.r.airflow"]
4599                             }
4600                     },
4601                     "schema": { "$ref": "#/definitions/RetrieveSchema" }
4602                 }
4603             },
4604             "post": {
4605                 "description": "",
4606                 "parameters": [
4607                     {"$ref": "#/parameters/interface-actuator"},
4608                     {
4609                         "name": "body",
4610                         "in": "body",
4611                         "required": true,
4612                         "schema": { "$ref": "#/definitions/UpdateSchema" }
4613                     }
4614                 ],
4615                 "responses": {
4616                     "200": {
4617                         "description": "",
4618                         "schema": { "$ref": "#/definitions/UpdateSchema" }
4619                     }
4620                 }
4621             }
4622         }
4623     },
4624     "parameters": {
4625         "interface-actuator" : {
4626             "in" : "query",
4627             "name" : "if",
4628             "type" : "string",
4629             "enum" : ["oic.if.a", "oic.if.baseline"]
4630         }
4631     },
4632     "definitions": {
4633         "RetrieveSchema" :
4634         {
4635             "properties": {
4636                 "automode": {
4637                     "description": "Auto mode status.",
4638                     "type": "integer",
4639                     "x-ocf-conversion": {
4640                         "x-from-ocf": [
4641                             "automode = ocf.automode",
4642                             "otherwise: automode = NotSupported(0xFF)"
4643                         ],
4644                         "x-ocf-alias": "oic.r.airflow",
4645                         "x-to-ocf": [
4646                             "if automode != NotSupported(0xFF)",
4647                             " ocf.automode = automode",
4648                             "else no mapping"
4649

```

```

4650     ]
4651   }
4652 },
4653   "fanspeedlevel": {
4654     "description": "Fan speed level. 0 = off.",
4655     "type": "integer",
4656     "x-ocf-conversion": {
4657       "x-from-ocf": [
4658         "fanspeedlevel = speed"
4659       ],
4660       "x-ocf-alias": "oic.r.airflow",
4661       "x-to-ocf": [
4662         "speed = fanspeedlevel"
4663       ]
4664     }
4665 },
4666   "maxfanspeedlevel": {
4667     "description": "Max level allowed for fan speed",
4668     "type": "integer",
4669     "x-ocf-conversion": {
4670       "x-from-ocf": [
4671         "maxfanspeedlevel = range[1]",
4672         "otherwise: maxfanspeedlevel = 100"
4673       ],
4674       "x-ocf-alias": "oic.r.airflow",
4675       "x-to-ocf": [
4676         "range[0] = 0",
4677         "range[1] = maxfanspeedlevel"
4678       ]
4679     }
4680   }
4681 },
4682   "required": [
4683     "fanspeedlevel",
4684     "maxfanspeedlevel",
4685     "automode"
4686   ],
4687   "type": "object"
4688 }
4689
4690 ,
4691 "UpdateSchema" :
4692 {
4693   "properties": {
4694     "automode": {
4695       "description": "Auto mode status.",
4696       "type": "integer",
4697       "x-ocf-conversion": {
4698         "x-from-ocf": [
4699           "automode = ocf.automode",
4700           "otherwise: automode = NotSupported(0xFF)"
4701         ],
4702         "x-ocf-alias": "oic.r.airflow",
4703         "x-to-ocf": [
4704           "if automode != NotSupported(0xFF)",
4705           " ocf.automode = automode",
4706           "else no mapping"
4707         ]
4708       }
4709     },
4710     "fanspeedlevel": {
4711       "description": "Fan speed level. 0 = off.",
4712       "type": "integer",
4713       "x-ocf-conversion": {
4714         "x-from-ocf": [
4715           "fanspeedlevel = speed"
4716         ],
4717         "x-ocf-alias": "oic.r.airflow",
4718         "x-to-ocf": [
4719           "speed = fanspeedlevel"
4720         ]

```

```

4721     }
4722   },
4723   "maxfanspeedlevel": {
4724     "description": "Max level allowed for fan speed",
4725     "type": "integer",
4726     "x-ocf-conversion": {
4727       "x-from-ocf": [
4728         "maxfanspeedlevel = range[1]",
4729         "otherwise: maxfanspeedlevel = 100"
4730       ],
4731       "x-ocf-alias": "oic.r.airflow",
4732       "x-to-ocf": [
4733         "range[0] = 0",
4734         "range[1] = maxfanspeedlevel"
4735       ]
4736     }
4737   }
4738 },
4739 "required": [
4740   "fanspeedlevel",
4741   "maxfanspeedlevel",
4742   "automode"
4743 ],
4744 "type": "object"
4745 }
4746
4747 }
4748 }
4749

```

4750 **A.9.5 Property Definition**

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
maxfanspeedlevel	oic.r.airflow	range[0] = 0 range[1] = maxfanspeedlevel	maxfanspeedlevel = range[1] otherwise: maxfanspeedlevel = 100	Max level allowed for fan speed
automode	oic.r.airflow	if automode != NotSupported(0xFF)) ocf.automode = automode else no mapping	automode = ocf.automode otherwise: automode = NotSupported(0xFF)	Auto mode status.
fanspeedlevel	oic.r.airflow	speed = fanspeedlevel	fanspeedlevel = speed	Fan speed level. 0 = off.

4751 **A.9.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/FanSpeedLevelResURI		get	post		

4752 **A.10 Heating Zone Mapping**

4753 **A.10.1 Introduction**

4754 This API defines the mapping between an instance of an AllJoyn HeatingZone interface and an
4755 OCF HeatingZoneCollection Resource.
4756 Each element in the array of heating zones within the AllJoyn HeatingZone interface maps to an
4757 instance of OCF HeatingZone, itself a link in an instance of an OCF HeatingZoneCollection.
4758 The mapping defined in the schema describes the population of the OCF HeatingZone Resource
4759 that constitutes the Resources that are contained in the collection.
4760

4761 **A.10.2 Example URI**

4762 /HeatingZoneResURI

4763 **A.10.3 Resource Type**

4764 The resource type (rt) is defined as: ['oic.r.heatingzonecollection'].

4765 **A.10.4 Swagger2.0 Definition**

```
4766 {
4767   "swagger": "2.0",
4768   "info": {
4769     "title": "Heating Zone Mapping",
4770     "version": "OCFv1.0.0-20170317",
4771     "license": {
4772       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
4773       "x-description": "Redistribution and use in source and binary forms, with or without
4774 modification, are permitted provided that the following conditions are met:\n      1.
4775 Redistributions of source code must retain the above copyright notice, this list of conditions and
4776 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
4777 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
4778 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
4779 Connectivity Foundation, INC. \AS IS\ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
4780 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
4781 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
4782 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
4783 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
4784 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND
4785 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
4786 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
4787 OF SUCH DAMAGE.\n"
4788     }
4789   },
4790   "schemes": ["http"],
4791   "consumes": ["application/json"],
4792   "produces": ["application/json"],
4793   "paths": {
4794     "/HeatingZoneResURI" : {
4795       "get": {
4796         "description": "This API defines the mapping between an instance of an AllJoyn HeatingZone
4797 interface and an OCF HeatingZoneCollection Resource.\nEach element in the array of heating zones
4798 within the AllJoyn HeatingZone interface maps to an instance of OCF HeatingZone, itself a link in
4799 an instance of an OCF HeatingZoneCollection.\nThe mapping defined in the schema describes the
4800 population of the OCF HeatingZone Resource that constitutes the Resources that are contained in the
4801 collection.\n",
4802         "parameters": [
4803           { "$ref": "#/parameters/interface-sensor" }
4804         ],
4805         "responses": {
4806           "200": {
4807             "description": "",
4808             "x-example":
4809               {
4810                 "rt": ["oic.r.heatingzonecollection"]
4811               }
4812           },
4813           "schema": { "$ref": "#/definitions/RetrieveSchema" }
4814         }
4815       }
4816     }
4817   },
4818 },
4819 "parameters": {
4820   "interface-sensor" : {
4821     "in" : "query",
4822     "name" : "if",
4823     "type" : "string",
4824     "enum" : ["oic.if.s", "oic.if.baseline"]
4825   }
4826 },
4827 "definitions": {
```

```

4828     "RetrieveSchema" :
4829     {
4830         "properties": {
4831             "heatinglevels": {
4832                 "description": "Current heating levels for each zone.",
4833                 "items": {
4834                     "type": "integer"
4835                 },
4836                 "type": "array",
4837                 "x-ocf-conversion": {
4838                     "x-from-ocf": [
4839                         "for x=0;x<numlinks(oic.r.heatingzonecollection): heatinglevels[x] =
4840 ocf.heatinglevel"
4841                     ],
4842                     "x-ocf-alias": "oic.r.heatingzone",
4843                     "x-to-ocf": [
4844                         "Instance of oic.r.heatingzone per array item ",
4845                         "for x=0, x<sizeof(heatinglevels): ocf.heatinglevel = maxheatinglevels[x]"
4846                     ]
4847                 }
4848             },
4849             "maxheatinglevels": {
4850                 "description": "Max heating levels for each zone",
4851                 "items": {
4852                     "type": "integer"
4853                 },
4854                 "type": "array",
4855                 "x-ocf-conversion": {
4856                     "x-from-ocf": [
4857                         "for x=0;x<numlinks(oic.r.heatingzonecollection): maxheatinglevels[x] =
4858 ocf.maxheatinglevel"
4859                     ],
4860                     "x-ocf-alias": "oic.r.heatingzone",
4861                     "x-to-ocf": [
4862                         "Instance of oic.r.heatingzone per array item ",
4863                         "for x=0, x<sizeof(maxheatinglevels): ocf.maxheatinglevel = maxheatinglevels[x]"
4864                     ]
4865                 }
4866             },
4867             "numberofheatingzones": {
4868                 "description": "Number of heating zones.",
4869                 "type": "integer",
4870                 "x-ocf-conversion": {
4871                     "x-from-ocf": [
4872                         "numberofheatingzones = number of links in the collection"
4873                     ],
4874                     "x-ocf-alias": "oic.r.heatingzonecollection",
4875                     "x-to-ocf": [
4876                         "number of links in the collection = numberofheatingzones"
4877                     ]
4878                 }
4879             }
4880         },
4881         "required": [
4882             "numberofheatingzones",
4883             "maxheatinglevels",
4884             "heatinglevels"
4885         ],
4886         "type": "object"
4887     }
4888 }
4889 }
4890 }
4891

```

4892 A.10.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Descri ption
---------------------------------	--------------	--------	----------	-----------------

heatinglevels	oic.r.heatingzone	Instance of oic.r.heatingzone per array item for x=0, x<sizeof(heatinglevels): ocf.heatinglevel = maxheatinglevels[x]	for x=0;x<numlinks(oic.r.heatingzonecollection): heatinglevels[x] = ocf.heatinglevel	Current heating levels for each zone.
numberofheatingzones	oic.r.heatingzonecollection	number of links in the collection = numberofheatingzones	numberofheatingzones = number of links in the collection	Number of heating zones.
maxheatinglevels	oic.r.heatingzone	Instance of oic.r.heatingzone per array item for x=0, x<sizeof(maxheatinglevels): ocf.maxheatinglevel = maxheatinglevels[x]	for x=0;x<numlinks(oic.r.heatingzonecollection): maxheatinglevels[x] = ocf.maxheatinglevel	Max heating levels for each zone

4893 **A.10.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/HeatingZoneResURI		get			

4894 **A.11 HVAC Fan Mode Mapping**

4895 **A.11.1 Introduction**

4896 This API defines the mapping between an instance of an AllJoyn HvacFanMode interface and an
 4897 OCF Mode Resource.
 4898

4899 **A.11.2 Example URI**

4900 /HvacFanModeResURI

4901 **A.11.3 Resource Type**

4902 The resource type (rt) is defined as: ['oic.r.mode'].

4903 **A.11.4 Swagger2.0 Definition**

```

4904 {
4905   "swagger": "2.0",
4906   "info": {
4907     "title": "HVAC Fan Mode Mapping",
4908     "version": "OCFv1.0.0-20170317",
4909     "license": {
4910       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
4911       "x-description": "Redistribution and use in source and binary forms, with or without
4912 modification, are permitted provided that the following conditions are met:\n      1.
4913 Redistributions of source code must retain the above copyright notice, this list of conditions and
4914 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
4915 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
4916 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
4917 Connectivity Foundation, INC. \\"AS IS\\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
4918 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
4919 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n\n      IN NO EVENT SHALL THE Open Connectivity
4920 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
4921 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

```

```

4922 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n          HOWEVER CAUSED AND
4923 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
4924 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
4925 OF SUCH DAMAGE.\n"
4926     }
4927   },
4928   "schemes": ["http"],
4929   "consumes": ["application/json"],
4930   "produces": ["application/json"],
4931   "paths": {
4932     "/HvacFanModeResURI" : {
4933       "get": {
4934         "description": "This API defines the mapping between an instance of an AllJoyn HvacFanMode
4935 interface and an OCF Mode Resource.\n",
4936         "parameters": [
4937           { "$ref": "#/parameters/interface-actuator" }
4938         ],
4939         "responses": {
4940           "200": {
4941             "description": "",
4942             "x-example":
4943               {
4944                 "rt": ["oic.r.mode"]
4945               }
4946             ,
4947             "schema": { "$ref": "#/definitions/RetrieveSchema" }
4948           }
4949         }
4950       },
4951       "post": {
4952         "description": "",
4953         "parameters": [
4954           { "$ref": "#/parameters/interface-actuator" },
4955           {
4956             "name": "body",
4957             "in": "body",
4958             "required": true,
4959             "schema": { "$ref": "#/definitions/UpdateSchema" }
4960           }
4961         ],
4962         "responses": {
4963           "200": {
4964             "description": "",
4965             "schema": { "$ref": "#/definitions/UpdateSchema" }
4966           }
4967         }
4968       }
4969     }
4970   },
4971   "parameters": {
4972     "interface-actuator" : {
4973       "in": "query",
4974       "name": "if",
4975       "type": "string",
4976       "enum": ["oic.if.a", "oic.if.baseline"]
4977     }
4978   },
4979   "definitions": {
4980     "RetrieveSchema" :
4981     {
4982       "properties": {
4983         "mode": {
4984           "description": "Current mode of device.",
4985           "type": "integer",
4986           "x-ocf-conversion": {
4987             "x-from-ocf": [
4988               "modearray = [Auto,Circulation,Continuous]",
4989               "mode = indexOf modeArray[ocf.mode[0]]"
4990             ],
4991             "x-ocf-alias": "oic.r.mode",
4992             "x-to-ocf": [

```

```

4993         "modearray = [Auto,Circulation,Continuous]",
4994         "ocf.mode[0] = modearray[mode]"
4995     ]
4996 }
4997 },
4998 "supportedmodes": {
4999     "description": "Array of supported modes",
5000     "items": {
5001         "type": "integer"
5002     },
5003     "type": "array",
5004     "x-ocf-conversion": {
5005         "x-from-ocf": [
5006             "modearray = [Auto,Circulation,Continuous]",
5007             "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
5008 modearray[ocf.supportedmodes[x]]"
5009         ],
5010         "x-ocf-alias": "oic.r.mode",
5011         "x-to-ocf": [
5012             "modearray = [Auto,Circulation,Continuous]",
5013             "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
5014 modearray[supportedmodes[x]]"
5015         ]
5016     }
5017 },
5018 },
5019 "required": [
5020     "mode",
5021     "supportedmodes"
5022 ],
5023 "type": "object"
5024 }
5025
5026
5027 "UpdateSchema" :
5028 {
5029     "properties": {
5030         "mode": {
5031             "description": "Current mode of device.",
5032             "type": "integer",
5033             "x-ocf-conversion": {
5034                 "x-from-ocf": [
5035                     "modearray = [Auto,Circulation,Continuous]",
5036                     "mode = indexof modeArray[ocf.mode[0]]"
5037                 ],
5038                 "x-ocf-alias": "oic.r.mode",
5039                 "x-to-ocf": [
5040                     "modearray = [Auto,Circulation,Continuous]",
5041                     "ocf.mode[0] = modearray[mode]"
5042                 ]
5043             }
5044         },
5045         "supportedmodes": {
5046             "description": "Array of supported modes",
5047             "items": {
5048                 "type": "integer"
5049             },
5050             "type": "array",
5051             "x-ocf-conversion": {
5052                 "x-from-ocf": [
5053                     "modearray = [Auto,Circulation,Continuous]",
5054                     "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
5055 modearray[ocf.supportedmodes[x]]"
5056                 ],
5057                 "x-ocf-alias": "oic.r.mode",
5058                 "x-to-ocf": [
5059                     "modearray = [Auto,Circulation,Continuous]",
5060                     "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
5061 modearray[supportedmodes[x]]"
5062                 ]
5063             }

```

```

5064     },
5065     },
5066     "required": [
5067         "mode",
5068         "supportedmodes"
5069     ],
5070     "type": "object"
5071 }
5072 }
5073 }
5074 }
5075 }

```

5076 **A.11.5 Property Definition**

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
supportedmodes	oic.r.mode	modearray = [Auto,Circulation,Continuous]for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] = modearray[supportedmodes[x]]	modearray = [Auto,Circulation,Continuous]for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof modearray[ocf.supportedmodes[x]]	Array of supported modes
mode	oic.r.mode	modearray = [Auto,Circulation,Continuous]ocf.mode[0] = modearray[mode]	modearray = [Auto,Circulation,Continuous]mode = indexof modeArray[ocf.mode[0]]	Current mode of device.

5077 **A.11.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/HvacFanModeResURI		get	post		

5078 **A.12 On Off Mapping**

5079 **A.12.1 Introduction**

5080 This API defines the mapping between an instance of an OCF Binary Switch Resource and the
5081 equivalent Interface set by AllJoyn. A discovered instance of a Binary Switch is always
5082 mapped to an Operation.OnOffStatus interface.
5083 A RETRIEVE on a Binary Switch maps to an action on an instance of an Operation.OnOffStatus
5084 Interface.

5085 An UPDATE on a Binary Switch maps to a method invocation on either Operation.OnControl or
5086 OffControl.

5087 value = true maps to Operation.OnControl
5088 value = false maps to Operation.OffControl
5089

5090 **A.12.2 Example URI**

5091 /OnOffResURI

5092 **A.12.3 Resource Type**

5093 The resource type (rt) is defined as: ['oic.r.switch.binary'].

5094 **A.12.4 Swagger2.0 Definition**

```

5095 {
5096     "swagger": "2.0",
5097     "info": {
5098         "title": "On Off Mapping",
5099         "version": "OCFv1.0.0-20170317",

```

```

5100     "license": {
5101         "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
5102         "x-description": "Redistribution and use in source and binary forms, with or without
5103 modification, are permitted provided that the following conditions are met:\n      1.
5104 Redistributions of source code must retain the above copyright notice, this list of conditions and
5105 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
5106 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
5107 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
5108 Connectivity Foundation, INC. \ "AS IS\ " AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
5109 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
5110 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
5111 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
5112 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
5113 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND
5114 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
5115 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
5116 OF SUCH DAMAGE.\n"
5117     }
5118 },
5119 "schemes": ["http"],
5120 "consumes": ["application/json"],
5121 "produces": ["application/json"],
5122 "paths": {
5123     "/OnOffResURI" : {
5124         "get": {
5125             "description": "This API defines the mapping between an instance of an OCF Binary Switch
5126 Resource and the\nequivalent Interface set by AllJoyn. A discovered instance of a Binary Switch is
5127 always\nmapped to an Operation.OnOffStatus interface.\nA RETRIEVE on a Binary Switch maps to an
5128 action on an instance of an Operation.OnOffStatus Interface.\nAn UPDATE on a Binary Switch maps to
5129 a method invocation on either Operation.OnControl or OffControl.\n value = true maps to
5130 Operation.OnControl\n value = false maps to Operation.OffControl\n",
5131             "parameters": [
5132                 {"$ref": "#/parameters/interface-all"}
5133             ],
5134             "responses": {
5135                 "200": {
5136                     "description": "",
5137                     "x-example":
5138                         {
5139                             "rt": ["oic.r.switch.binary"]
5140                         }
5141                 },
5142                 "schema": { "$ref": "#/definitions/RetrieveSchema" }
5143             }
5144         }
5145     },
5146     "post": {
5147         "description": "",
5148         "parameters": [
5149             {"$ref": "#/parameters/interface-actuator"},
5150             {
5151                 "name": "body",
5152                 "in": "body",
5153                 "required": true,
5154                 "schema": { "$ref": "#/definitions/UpdateSchema" }
5155             }
5156         ],
5157         "responses": {
5158             "200": {
5159                 "description": "",
5160                 "schema": { "$ref": "#/definitions/UpdateSchema" }
5161             }
5162         }
5163     }
5164 }
5165 },
5166 "parameters": {
5167     "interface-actuator" : {
5168         "in" : "query",
5169         "name" : "if",
5170         "type" : "string",

```

```

5171     "enum" : ["oic.if.a", "oic.if.baseline"]
5172   },
5173   "interface-all" : {
5174     "in" : "query",
5175     "name" : "if",
5176     "type" : "string",
5177     "enum" : ["oic.if.s", "oic.if.a", "oic.if.baseline"]
5178   }
5179 },
5180 "definitions": {
5181   "RetrieveSchema" :
5182     {
5183     "properties": {
5184       "onoff": {
5185         "description": "On/Off status of the device",
5186         "type": "boolean",
5187         "x-ocf-conversion": {
5188           "x-from-ocf": [
5189             "onoff = value"
5190           ],
5191           "x-ocf-alias": "oic.r.switch.binary",
5192           "x-to-ocf": [
5193             "value = onoff"
5194           ]
5195         }
5196       }
5197     },
5198     "required": [
5199       "onoff"
5200     ],
5201     "type": "object"
5202   }
5203
5204   ,
5205   "UpdateSchema" :
5206     {
5207     "properties": {
5208       "switchon": {
5209         "description": "Turn on the device",
5210         "format": "method",
5211         "type": "string",
5212         "x-ocf-conversion": {
5213           "x-ocf-alias": "oic.r.switch.binary",
5214           "x-to-ocf": [
5215             "value = true"
5216           ]
5217         }
5218       }
5219     },
5220     "type": "object"
5221   }
5222 }
5223 }
5224 }
5225

```

5226 **A.12.5 Property Definition**

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
onoff	oic.r.switch.binary	value = onoff	onoff = value	On/Off status of the device

5227 **A.12.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/OnOffResURI		get	post		

5228 **A.13 Oven Cycle Phase Mapping**

5229 **A.13.1 Introduction**

5230 This API defines the mapping between an instance of an AllJoyn OvenCyclePhase interface and
5231 the OCF OperationalState Resource.
5232 OvenCyclePhase cyclephase Property pre-defines values 0x00-0x7F, 0x80-0xFF is for vendor
5233 specific values. The mapping defined herein covers only Spec defined values.
5234 Any vendor defined value shall be represented in OCF using the x.<organization> syntax for a
5235 vendor defined Property.
5236 The AllJoyn interface also supports a Method, GetVendorPhasesDescription; this is handled in
5237 OCF using an instance of oic.r.actuator within an oic.r.action collection.
5238 Please see Section 8 of the Mapping Specification for specifics.
5239

5240 **A.13.2 Example URI**

5241 /OvenCyclePhaseResURI

5242 **A.13.3 Resource Type**

5243 The resource type (rt) is defined as: ['oic.r.operationalstate'].

5244 **A.13.4 Swagger2.0 Definition**

```
5245 {  
5246   "swagger": "2.0",  
5247   "info": {  
5248     "title": "Oven Cycle Phase Mapping",  
5249     "version": "OCFv1.0.0-20170317",  
5250     "license": {  
5251       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",  
5252       "x-description": "Redistribution and use in source and binary forms, with or without  
5253 modification, are permitted provided that the following conditions are met:\n      1.  
5254 Redistributions of source code must retain the above copyright notice, this list of conditions and  
5255 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above  
5256 copyright notice, this list of conditions and the following disclaimer in the documentation and/or  
5257 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open  
5258 Connectivity Foundation, INC. \\\nAS IS\\n AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT  
5259 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR  
5260 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n\n      IN NO EVENT SHALL THE Open Connectivity  
5261 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,  
5262 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
5263 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n\n      HOWEVER CAUSED AND  
5264 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR  
5265 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY  
5266 OF SUCH DAMAGE.\n\n    }  
5267   },  
5268   "schemes": ["http"],  
5269   "consumes": ["application/json"],  
5270   "produces": ["application/json"],  
5271   "paths": {  
5272     "/OvenCyclePhaseResURI" : {  
5273       "get": {  
5274         "description": "This API defines the mapping between an instance of an AllJoyn  
5275 OvenCyclePhase interface and the OCF OperationalState Resource.\nOvenCyclePhase cyclephase Property  
5276 pre-defines values 0x00-0x7F, 0x80-0xFF is for vendor specific values. The mapping defined herein  
5277 covers only Spec defined values.\nAny vendor defined value shall be represented in OCF using the  
5278 x.<organization> syntax for a vendor defined Property.\nThe AllJoyn interface also supports a  
5279 Method, GetVendorPhasesDescription; this is handled in OCF using an instance of oic.r.actuator  
5280 within an oic.r.action collection.\nPlease see Section 8 of the Mapping Specification for  
5281 specifics.\n",  
5282         "parameters": [  
5283           {  
5284             "$ref": "#/parameters/interface-sensor"  
5285           }  
5286         ],  
5287         "responses": {  
5288           "200": {  
5289             "description": ""  
5290           }  
5291         }  
5292       }  
5293     }  
5294   }  
5295 }
```

```

5289         "x-example":
5290         {
5291             "rt":      ["oic.r.operationalstate"]
5292         }
5293     },
5294     "schema": { "$ref": "#/definitions/RetrieveSchema" }
5295 }
5296 }
5297 }
5298 }
5299 },
5300 "parameters": {
5301     "interface-sensor" : {
5302         "in" : "query",
5303         "name" : "if",
5304         "type" : "string",
5305         "enum" : ["oic.if.s", "oic.if.baseline"]
5306     }
5307 },
5308 "definitions": {
5309     "RetrieveSchema" :
5310     {
5311         "properties": {
5312             "cyclephase": {
5313                 "description": "Current phase of the operational cycle",
5314                 "type": "integer",
5315                 "x-ocf-conversion": {
5316                     "x-from-ocf": [
5317                         "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
5318                         "cyclephase = indexof statearray[currentmachinestate[0]]"
5319                     ],
5320                     "x-ocf-alias": "oic.r.operationalstate",
5321                     "x-to-ocf": [
5322                         "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
5323                         "currentmachinestate = phasearray[cyclephase]"
5324                     ]
5325                 }
5326             },
5327             "getvendorphasesdescription": {
5328                 "description": "Get cycle phases description",
5329                 "x-ocf-conversion": {
5330                     "x-ocf-alias": "oic.r.action"
5331                 },
5332                 "x-ocf-type": "method"
5333             },
5334             "supportedcyclephases": {
5335                 "description": "Array of cycle phases supported by the Appliance.",
5336                 "items": {
5337                     "type": "integer"
5338                 },
5339                 "type": "array",
5340                 "x-ocf-conversion": {
5341                     "x-from-ocf": [
5342                         "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
5343                         "for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof
5344 phasearray[machinestates[x]]"
5345                     ],
5346                     "x-ocf-alias": "oic.r.operationalstate",
5347                     "x-to-ocf": [
5348                         "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
5349                         "for x=0, x < sizeof(supportedcyclephases): machinestates[x] =
5350 phasearray[supportedcyclephases[x]]"
5351                     ]
5352                 }
5353             }
5354         },
5355         "required": [
5356             "cyclephase",
5357             "supportedcyclephases"
5358         ],
5359         "type": "object"

```

5360 }
 5361 }
 5362 }
 5363 }
 5364 }

5365 **A.13.5 Property Definition**

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
supportedcyclephases	oic.r.operationalstate	phasearray = [Unavailable,Preheating,Cooking,Cleaning]for x=0, x < sizeof(supportedcyclephases): machinestates[x] = phasearray[supportedcyclephases[x]]	phasearray = [Unavailable,Preheating,Cooking,Cleaning]for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof phasearray[machinestates[x]]	Array of cycle phases supported by the Appliance.
cyclephase	oic.r.operationalstate	phasearray = [Unavailable,Preheating,Cooking,Cleaning]currentmachinestate = phasearray[cyclephase]	phasearray = [Unavailable,Preheating,Cooking,Cleaning]cyclephase = indexof statearray[currentmachinestate[0]]	Current phase of the operational cycle
getvendorphasedescription	oic.r.action			Get cycle phases description

5366 **A.13.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/OvenCyclePhaseResURI		get			

5367 **A.14 Target Humidity Mapping**

5368 **A.14.1 Introduction**

5369 This API defines the mapping between an instance of an AllJoyn TargetHumidity Interface and the
 5370 OCF Resource Equivalent.
 5371 A POST on a Humidity Sensor maps to an action on an instance of an Environment.TargetHumidity
 5372 Interface.
 5373

5374 **A.14.2 Example URI**

5375 /TargetHumidityResURI

5376 **A.14.3 Resource Type**

5377 The resource type (rt) is defined as: ['oic.r.humidity', 'oic.r.selectablelevels'].

5378 A.14.4 Swagger2.0 Definition

```
5379 {
5380   "swagger": "2.0",
5381   "info": {
5382     "title": "Target Humidity Mapping",
5383     "version": "OCFv1.0.0-20170317",
5384     "license": {
5385       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
5386       "x-description": "Redistribution and use in source and binary forms, with or without
5387 modification, are permitted provided that the following conditions are met:\n      1.
5388 Redistributions of source code must retain the above copyright notice, this list of conditions and
5389 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
5390 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
5391 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
5392 Connectivity Foundation, INC. \\"AS IS\\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
5393 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
5394 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
5395 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
5396 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
5397 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND
5398 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
5399 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
5400 OF SUCH DAMAGE.\n"
5401     }
5402   },
5403   "schemes": ["http"],
5404   "consumes": ["application/json"],
5405   "produces": ["application/json"],
5406   "paths": {
5407     "/TargetHumidityResURI" : {
5408       "get": {
5409         "description": "This API defines the mapping between an instance of an AllJoyn
5410 TargetHumidity Interface and the OCF Resource Equivalent.\nA POST on a Humidity Sensor maps to an
5411 action on an instance of an Environment.TargetHumidity Interface.\n",
5412         "parameters": [
5413           { "$ref": "#/parameters/interface-actuator" }
5414         ],
5415         "responses": {
5416           "200": {
5417             "description": "",
5418             "x-example":
5419               {
5420                 "rt": ["oic.r.humidity","oic.r.selectablelevels"]
5421               }
5422             ,
5423             "schema": { "$ref": "#/definitions/RetrieveSchema" }
5424           }
5425         }
5426       },
5427       "post": {
5428         "description": "",
5429         "parameters": [
5430           { "$ref": "#/parameters/interface-actuator" },
5431           {
5432             "name": "body",
5433             "in": "body",
5434             "required": true,
5435             "schema": { "$ref": "#/definitions/UpdateSchema" }
5436           }
5437         ],
5438         "responses": {
5439           "200": {
5440             "description": "",
5441             "x-example":
5442               {
5443                 "rt": ["oic.r.humidity","oic.r.selectablelevels"]
5444               }
5445             ,
5446             "schema": { "$ref": "#/definitions/UpdateSchema" }
5447           }
5448         }
5449       }
5450     }
5451   }
5452 }
```

```

5448     }
5449   }
5450 }
5451 },
5452 "parameters": {
5453   "interface-actuator" : {
5454     "in" : "query",
5455     "name" : "if",
5456     "type" : "string",
5457     "enum" : ["oic.if.a", "oic.if.baseline"]
5458   }
5459 },
5460 "definitions": {
5461   "RetrieveSchema" :
5462     {
5463     "properties": {
5464       "maxvalue": {
5465         "type": "number",
5466         "x-ocf-conversion": {
5467           "x-from-ocf": [
5468             "maxvalue = range[1]",
5469             "otherwise: maxvalue = 100"
5470           ],
5471           "x-ocf-alias": "oic.r.humidity",
5472           "x-to-ocf": [
5473             "range[1] = maxvalue"
5474           ]
5475         }
5476       },
5477       "minvalue": {
5478         "type": "number",
5479         "x-ocf-conversion": {
5480           "x-from-ocf": [
5481             "minvalue = range[0]",
5482             "otherwise: minvalue = 0"
5483           ],
5484           "x-ocf-alias": "oic.r.humidity",
5485           "x-to-ocf": [
5486             "range[0] = minvalue"
5487           ]
5488         }
5489       },
5490       "selectablehumiditylevels": {
5491         "items": {
5492           "type": "number"
5493         },
5494         "type": "array",
5495         "x-ocf-conversion": {
5496           "x-from-ocf": [
5497             "selectablehumiditylevels[] = availablelevels[]"
5498           ],
5499           "x-ocf-alias": "oic.r.selectablelevels",
5500           "x-to-ocf": [
5501             "availablelevels[] = selectablehumiditylevels[]"
5502           ]
5503         }
5504       },
5505       "stepvalue": {
5506         "type": "number",
5507         "x-ocf-conversion": {
5508           "x-from-ocf": [
5509             "stepvalue = step",
5510             "otherwise: stepvalue = 1"
5511           ],
5512           "x-ocf-alias": "oic.r.humidity",
5513           "x-to-ocf": [
5514             "step = stepvalue"
5515           ]
5516         }
5517       },
5518       "targetvalue": {

```

```

5519         "description": "Measured value",
5520         "type": "number",
5521         "x-ocf-conversion": {
5522             "x-from-ocf": [
5523                 "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
5524                 "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
5525             ],
5526             "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
5527             "x-to-ocf": [
5528                 "if minvalue != maxvalue, ocf.desiredhumidity = targetvalue;ocf.targetlevel =
5529 selectablehumiditylevels[0].",
5530                 "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
5531             ]
5532         }
5533     },
5534 },
5535 "required": [
5536     "targetvalue",
5537     "minvalue",
5538     "maxvalue",
5539     "stepvalue",
5540     "selectablehumiditylevels"
5541 ],
5542 "type": "object"
5543 }
5544
5545 ,
5546 "UpdateSchema" :
5547 {
5548     "properties": {
5549         "maxvalue": {
5550             "type": "number",
5551             "x-ocf-conversion": {
5552                 "x-from-ocf": [
5553                     "maxvalue = range[1]",
5554                     "otherwise: maxvalue = 100"
5555                 ],
5556                 "x-ocf-alias": "oic.r.humidity",
5557                 "x-to-ocf": [
5558                     "range[1] = maxvalue"
5559                 ]
5560             }
5561         },
5562         "minvalue": {
5563             "type": "number",
5564             "x-ocf-conversion": {
5565                 "x-from-ocf": [
5566                     "minvalue = range[0]",
5567                     "otherwise: minvalue = 0"
5568                 ],
5569                 "x-ocf-alias": "oic.r.humidity",
5570                 "x-to-ocf": [
5571                     "range[0] = minvalue"
5572                 ]
5573             }
5574         },
5575         "selectablehumiditylevels": {
5576             "items": {
5577                 "type": "number"
5578             },
5579             "type": "array",
5580             "x-ocf-conversion": {
5581                 "x-from-ocf": [
5582                     "selectablehumiditylevels[] = availablelevels[]"
5583                 ],
5584                 "x-ocf-alias": "oic.r.selectablelevels",
5585                 "x-to-ocf": [
5586                     "availablelevels[] = selectablehumiditylevels[]"
5587                 ]
5588             }
5589         },

```

```

5590     "stepvalue": {
5591       "type": "number",
5592       "x-ocf-conversion": {
5593         "x-from-ocf": [
5594           "stepvalue = step",
5595           "otherwise: stepvalue = 1"
5596         ],
5597         "x-ocf-alias": "oic.r.humidity",
5598         "x-to-ocf": [
5599           "step = stepvalue"
5600         ]
5601       }
5602     },
5603     "targetvalue": {
5604       "description": "Measured value",
5605       "type": "number",
5606       "x-ocf-conversion": {
5607         "x-from-ocf": [
5608           "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
5609           "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
5610         ],
5611         "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
5612         "x-to-ocf": [
5613           "if minvalue != maxvalue, ocf.desiredhumidity = targetvalue;ocf.targetlevel =
5614 selectablehumiditylevels[0].",
5615           "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
5616         ]
5617       }
5618     }
5619   },
5620   "required": [
5621     "targetvalue",
5622     "minvalue",
5623     "maxvalue",
5624     "stepvalue",
5625     "selectablehumiditylevels"
5626   ],
5627   "type": "object"
5628 }
5629 }
5630 }
5631 }
5632 }

```

5633 A.14.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
stepvalue	oic.r.humidity	step = stepvalue	stepvalue = stepotherwise: stepvalue = 1	
targetvalue	oic.r.humidity,oic.r.selectablelevels	if minvalue != maxvalue, ocf.desiredhumidity = targetvalue;ocf.targ etlevel = selectablehumidityl evels[0].if minvalue == maxvalue, ocf.targetlevel = targetvalue.	if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.i f x-ocf-alias == oic.r.selectablele vels, targetvalue = targetlevel.	Measur ed value
maxvalue	oic.r.humidity	range[1] maxvalue	maxvalue = range[1]otherwis e: maxvalue = 100	

selectablehumiditylevels	oic.r.selectablelevels	availablelevels[] = selectablehumiditylevels[]	selectablehumiditylevels[] = availablelevels[]	
minvalue	oic.r.humidity	range[0] = minvalue	minvalue = range[0]otherwise: minvalue = 0	

5634 **A.14.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/TargetHumidityResURI		get	post		

5635 **A.15 Target Temperature Mapping**

5636 **A.15.1 Introduction**

5637 This API defines the mapping between an instance of an OCF Temperature which exposes only a
5638 sensor interface
5639 and the AllJoyn Current Temperature interface.
5640 A RETRIEVE on a Temperature Sensor maps to an action on an instance of an
5641 Environment.CurrentTemperature Interface.
5642

5643 **A.15.2 Example URI**

5644 /TargetTemperatureResURI

5645 **A.15.3 Resource Type**

5646 The resource type (rt) is defined as: ['oic.r.temperature'].

5647 **A.15.4 Swagger2.0 Definition**

```

5648 {
5649   "swagger": "2.0",
5650   "info": {
5651     "title": "Target Temperature Mapping",
5652     "version": "OCFv1.0.0-20170317",
5653     "license": {
5654       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
5655       "x-description": "Redistribution and use in source and binary forms, with or without
5656 modification, are permitted provided that the following conditions are met:\n      1.
5657 Redistributions of source code must retain the above copyright notice, this list of conditions and
5658 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
5659 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
5660 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
5661 Connectivity Foundation, INC. \AS IS\ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
5662 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
5663 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
5664 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
5665 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
5666 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND
5667 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
5668 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
5669 OF SUCH DAMAGE.\n"
5670     }
5671   },
5672   "schemes": ["http"],
5673   "consumes": ["application/json"],
5674   "produces": ["application/json"],
5675   "paths": {
5676     "/TargetTemperatureResURI" : {
5677       "get": {
5678         "description": "This API defines the mapping between an instance of an OCF Temperature
5679 which exposes only a sensor interface\nand the AllJoyn Current Temperature interface.\nA RETRIEVE
5680 on a Temperature Sensor maps to an action on an instance of an Environment.CurrentTemperature
5681 Interface.\n",
5682         "parameters": [

```



```

5683     {"$ref": "#/parameters/interface-actuator"}
5684   ],
5685   "responses": {
5686     "200": {
5687       "description": "",
5688       "x-example":
5689         {
5690           "rt": ["oic.r.temperature"]
5691         }
5692     },
5693     "schema": {"$ref": "#/definitions/RetrieveSchema"}
5694   }
5695 },
5696 ],
5697 "post": {
5698   "description": "",
5699   "parameters": [
5700     {"$ref": "#/parameters/interface-actuator"},
5701     {
5702       "name": "body",
5703       "in": "body",
5704       "required": true,
5705       "schema": {"$ref": "#/definitions/UpdateSchema"}
5706     }
5707   ],
5708   "responses": {
5709     "200": {
5710       "description": "",
5711       "schema": {"$ref": "#/definitions/UpdateSchema"}
5712     }
5713   }
5714 }
5715 },
5716 ],
5717 "parameters": {
5718   "interface-actuator": {
5719     "in": "query",
5720     "name": "if",
5721     "type": "string",
5722     "enum": ["oic.if.a", "oic.if.baseline"]
5723   }
5724 },
5725 "definitions": {
5726   "RetrieveSchema": {
5727     {
5728       "properties": {
5729         "maxvalue": {
5730           "type": "number",
5731           "x-ocf-conversion": {
5732             "x-from-ocf": [
5733               "maxvalue = range[1]",
5734               "otherwise: maxvalue = MAXINT"
5735             ],
5736             "x-ocf-alias": "oic.r.temperature",
5737             "x-to-ocf": [
5738               "range[1] = maxvalue"
5739             ]
5740           }
5741         },
5742         "minvalue": {
5743           "type": "number",
5744           "x-ocf-conversion": {
5745             "x-from-ocf": [
5746               "minvalue = range[0]",
5747               "otherwise: minvalue = -MAXINT"
5748             ],
5749             "x-ocf-alias": "oic.r.temperature",
5750             "x-to-ocf": [
5751               "range[0] = minvalue"
5752             ]
5753           }
5754       }
5755     }
5756   }
5757 }

```

```

5754     },
5755     "step": {
5756         "type": "number",
5757         "x-ocf-conversion": {
5758             "x-from-ocf": [
5759                 "step = ocf.step",
5760                 "otherwise: step = undefined (0x00)"
5761             ],
5762             "x-ocf-alias": "oic.r.temperature",
5763             "x-to-ocf": [
5764                 "ocf.step = step"
5765             ]
5766         }
5767     },
5768     "targetvalue": {
5769         "description": "Measured value",
5770         "type": "number",
5771         "x-ocf-conversion": {
5772             "x-from-ocf": {
5773                 "oneOf": [
5774                     {
5775                         "properties": {
5776                             "enum": [
5777                                 "C"
5778                             ],
5779                             "units": "string"
5780                         },
5781                         "x-from-ocf": [
5782                             "targetvalue = temperature"
5783                         ]
5784                     },
5785                     {
5786                         "properties": {
5787                             "enum": [
5788                                 "F"
5789                             ],
5790                             "units": "string"
5791                         },
5792                         "x-from-ocf": [
5793                             "targetvalue = (temperature-32)*5/9"
5794                         ]
5795                     },
5796                     {
5797                         "properties": {
5798                             "enum": [
5799                                 "K"
5800                             ],
5801                             "units": "string"
5802                         },
5803                         "x-from-ocf": [
5804                             "targetvalue = temperature-273.15"
5805                         ]
5806                     }
5807                 ]
5808             },
5809             "x-ocf-alias": "oic.r.temperature",
5810             "x-to-ocf": [
5811                 "temperature = targetvalue",
5812                 "units = C"
5813             ]
5814         }
5815     },
5816     "required": [
5817         "targetvalue",
5818         "minvalue",
5819         "maxvalue",
5820         "step"
5821     ],
5822     "type": "object"
5823 }
5824

```

```

5825
5826
5827 "UpdateSchema" :
5828 {
5829   "properties": {
5830     "maxvalue": {
5831       "type": "number",
5832       "x-ocf-conversion": {
5833         "x-from-ocf": [
5834           "maxvalue = range[1]",
5835           "otherwise: maxvalue = MAXINT"
5836         ],
5837         "x-ocf-alias": "oic.r.temperature",
5838         "x-to-ocf": [
5839           "range[1] = maxvalue"
5840         ]
5841       }
5842     },
5843     "minvalue": {
5844       "type": "number",
5845       "x-ocf-conversion": {
5846         "x-from-ocf": [
5847           "minvalue = range[0]",
5848           "otherwise: minvalue = -MAXINT"
5849         ],
5850         "x-ocf-alias": "oic.r.temperature",
5851         "x-to-ocf": [
5852           "range[0] = minvalue"
5853         ]
5854       }
5855     },
5856     "step": {
5857       "type": "number",
5858       "x-ocf-conversion": {
5859         "x-from-ocf": [
5860           "step = ocf.step",
5861           "otherwise: step = undefined (0x00)"
5862         ],
5863         "x-ocf-alias": "oic.r.temperature",
5864         "x-to-ocf": [
5865           "ocf.step = step"
5866         ]
5867       }
5868     },
5869     "targetvalue": {
5870       "description": "Measured value",
5871       "type": "number",
5872       "x-ocf-conversion": {
5873         "x-from-ocf": {
5874           "oneOf": [
5875             {
5876               "properties": {
5877                 "enum": [
5878                   "C"
5879                 ],
5880                 "units": "string"
5881             },
5882             "x-from-ocf": [
5883               "targetvalue = temperature"
5884             ]
5885           },
5886           {
5887             "properties": {
5888               "enum": [
5889                 "F"
5890               ],
5891               "units": "string"
5892             },
5893             "x-from-ocf": [
5894               "targetvalue = (temperature-32)*5/9"
5895             ]
5896           }
5897         ]
5898       }
5899     }
5900   }
5901 }

```

```

5896     },
5897     {
5898         "properties": {
5899             "enum": [
5900                 "K"
5901             ],
5902             "units": "string"
5903         },
5904         "x-from-ocf": [
5905             "targetvalue = temperature-273.15"
5906         ]
5907     }
5908 ],
5909 },
5910 "x-ocf-alias": "oic.r.temperature",
5911 "x-to-ocf": [
5912     "temperature = targetvalue",
5913     "units = C"
5914 ]
5915 }
5916 },
5917 },
5918 "required": [
5919     "targetvalue",
5920     "minvalue",
5921     "maxvalue",
5922     "step"
5923 ],
5924 "type": "object"
5925 }
5926 }
5927 }
5928 }
5929 }

```

5930 **A.15.5 Property Definition**

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
targetvalue	oic.r.temperature	temperature = targetvalueunits = C	oneOf	Measured value
step	oic.r.temperature	ocf.step = step	step = ocf.stepotherwise: step = undefined (0x00)	
maxvalue	oic.r.temperature	range[1] = maxvalue	maxvalue = range[1]otherwise: maxvalue = MAXINT	
minvalue	oic.r.temperature	range[0] = minvalue	minvalue = range[0]otherwise: minvalue = - MAXINT	

5931 **A.15.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/TargetTemperatureResURI		get	post		

5932
5933