

OCF Resource to AllJoyn Interface Mapping Specification

VERSION 1.0.0 | June 2017



OPEN CONNECTIVITY
FOUNDATION™

CONTACT admin@openconnectivity.org

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

Legal Disclaimer

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

CONTENTS

24	1	Scope	10
25	2	Normative references	10
26	3	Terms, definitions symbols and abbreviations	10
27	3.1	Terms and definitions	10
28	3.2	Symbols and abbreviations	10
29	3.3	Conventions	11
30	4	Document conventions and organization	11
31	4.1	Notation.....	11
32	4.2	Data types	12
33	5	Theory of Operation	12
34	5.1	Interworking Approach.....	12
35	5.2	Mapping Syntax.....	12
36	5.2.1	General.....	12
37	5.2.2	Value Assignment	12
38	5.2.3	Property Naming	12
39	5.2.4	Arrays	12
40	5.2.5	Default Mapping	12
41	5.2.6	Conditional Mapping.....	12
42	5.2.7	Loops.....	13
43	5.2.8	Method Invocation	13
44	6	Device Type Mapping.....	13
45	6.1	Introduction	13
46	6.2	AllJoyn Device Types to OCF Device Types	13
47	6.3	OCF Device Types with no AllJoyn Equivalent.....	14
48	7	Resource to Interface Equivalence	15
49	7.1	Introduction	15
50	7.2	AllJoyn Interfaces to OCF Resources	15
51	8	Detailed Mapping APIs	18
52	8.1	Air Quality Mapping	19
53	8.1.1	Introduction	19
54	8.1.2	Example URI	19
55	8.1.3	Resource Type	19
56	8.1.4	RAML Definition	19
57	8.1.5	Property Definition	21
58	8.1.6	CRUDN behavior.....	22
59	8.2	Air Quality Level Mapping.....	22
60	8.2.1	Introduction	22
61	8.2.2	Example URI	22
62	8.2.3	Resource Type	22

63	8.2.4	RAML Definition	22
64	8.2.5	Property Definition	24
65	8.2.6	CRUDN behavior	25
66	8.3	Current Humidity Mapping	25
67	8.3.1	Introduction	25
68	8.3.2	Example URI	25
69	8.3.3	Resource Type	25
70	8.3.4	RAML Definition	25
71	8.3.5	Property Definition	26
72	8.3.6	CRUDN behavior	26
73	8.4	Current Temperature Mapping	27
74	8.4.1	Introduction	27
75	8.4.2	Example URI	27
76	8.4.3	Resource Type	27
77	8.4.4	RAML Definition	27
78	8.4.5	Property Definition	29
79	8.4.6	CRUDN behavior	29
80	8.5	Target Humidity Mapping	29
81	8.5.1	Introduction	29
82	8.5.2	Example URI	29
83	8.5.3	Resource Type	29
84	8.5.4	RAML Definition	29
85	8.5.5	Property Definition	34
86	8.5.6	CRUDN behavior	34
87	8.6	Target Temperature Mapping	34
88	8.6.1	Introduction	34
89	8.6.2	Example URI	34
90	8.6.3	Resource Type	35
91	8.6.4	RAML Definition	35
92	8.6.5	Property Definition	40
93	8.6.6	CRUDN behavior	40
94	8.7	Audio Volume Mapping	40
95	8.7.1	Introduction	40
96	8.7.2	Example URI	40
97	8.7.3	Resource Type	40
98	8.7.4	RAML Definition	40
99	8.7.5	Property Definition	43
100	8.7.6	CRUDN behavior	44
101	8.8	Climate Control Mode Mapping	44
102	8.8.1	Introduction	44
103	8.8.2	Example URI	44
104	8.8.3	Resource Type	44
105	8.8.4	RAML Definition	44
106	8.8.5	Property Definition	48

107	8.8.6	CRUDN behavior	48
108	8.9	Closed Status Mapping	48
109	8.9.1	Introduction	48
110	8.9.2	Example URI	48
111	8.9.3	Resource Type	49
112	8.9.4	RAML Definition	49
113	8.9.5	Property Definition	50
114	8.9.6	CRUDN behavior	50
115	8.10	Cycle Control Mapping	50
116	8.10.1	Introduction	50
117	8.10.2	Example URI	50
118	8.10.3	Resource Type	50
119	8.10.4	RAML Definition	50
120	8.10.5	Property Definition	52
121	8.10.6	CRUDN behavior	52
122	8.11	Fan Speed Level Mapping	52
123	8.11.1	Introduction	52
124	8.11.2	Example URI	52
125	8.11.3	Resource Type	52
126	8.11.4	RAML Definition	52
127	8.11.5	Property Definition	56
128	8.11.6	CRUDN behavior	56
129	8.12	Heating Zone Mapping	56
130	8.12.1	Introduction	56
131	8.12.2	Example URI	56
132	8.12.3	Resource Type	56
133	8.12.4	RAML Definition	57
134	8.12.5	Property Definition	58
135	8.12.6	CRUDN behavior	59
136	8.13	HVAC Fan Mode Mapping	59
137	8.13.1	Introduction	59
138	8.13.2	Example URI	59
139	8.13.3	Resource Type	59
140	8.13.4	RAML Definition	59
141	8.13.5	Property Definition	62
142	8.13.6	CRUDN behavior	62
143	8.14	On Off Mapping	62
144	8.14.1	Introduction	62
145	8.14.2	Example URI	63
146	8.14.3	Resource Type	63
147	8.14.4	RAML Definition	63
148	8.14.5	Property Definition	66
149	8.14.6	CRUDN behavior	66
150	8.15	Oven Cycle Phase Mapping	67

151	8.15.1	Introduction	67
152	8.15.2	Example URI	67
153	8.15.3	Resource Type	67
154	8.15.4	RAML Definition	67
155	8.15.5	Property Definition	68
156	8.15.6	CRUDN behavior	69
157		Annex A Swagger2.0 (Informative)	70
158	A.1	Audio Volume Mapping	70
159	A.1.1	Introduction	70
160	A.1.2	Example URI	70
161	A.1.3	Resource Type	70
162	A.1.4	Swagger2.0 Definition	70
163	A.1.5	Property Definition	72
164	A.1.6	CRUDN behavior	73
165	A.2	Climate Control Mode Mapping	73
166	A.2.1	Introduction	73
167	A.2.2	Example URI	73
168	A.2.3	Resource Type	73
169	A.2.4	Swagger2.0 Definition	73
170	A.2.5	Property Definition	76
171	A.2.6	CRUDN behavior	76
172	A.3	Closed Status Mapping	77
173	A.3.1	Introduction	77
174	A.3.2	Example URI	77
175	A.3.3	Resource Type	77
176	A.3.4	Swagger2.0 Definition	77
177	A.3.5	Property Definition	78
178	A.3.6	CRUDN behavior	78
179	A.4	Air Quality Mapping	78
180	A.4.1	Introduction	78
181	A.4.2	Example URI	79
182	A.4.3	Resource Type	79
183	A.4.4	Swagger2.0 Definition	79
184	A.4.5	Property Definition	81
185	A.4.6	CRUDN behavior	81
186	A.5	Air Quality Level Mapping	81
187	A.5.1	Introduction	81
188	A.5.2	Example URI	82
189	A.5.3	Resource Type	82
190	A.5.4	Swagger2.0 Definition	82
191	A.5.5	Property Definition	84
192	A.5.6	CRUDN behavior	85
193	A.6	Current Humidity Mapping	85
194	A.6.1	Introduction	85

195	A.6.2	Example URI	85
196	A.6.3	Resource Type	85
197	A.6.4	Swagger2.0 Definition	85
198	A.6.5	Property Definition	86
199	A.6.6	CRUDN behavior	87
200	A.7	Current Temperature Mapping	87
201	A.7.1	Introduction	87
202	A.7.2	Example URI	87
203	A.7.3	Resource Type	87
204	A.7.4	Swagger2.0 Definition	87
205	A.7.5	Property Definition	89
206	A.7.6	CRUDN behavior	89
207	A.8	Cycle Control Mapping	89
208	A.8.1	Introduction	89
209	A.8.2	Example URI	89
210	A.8.3	Resource Type	90
211	A.8.4	Swagger2.0 Definition	90
212	A.8.5	Property Definition	91
213	A.8.6	CRUDN behavior	92
214	A.9	Fan Speed Level Mapping	92
215	A.9.1	Introduction	92
216	A.9.2	Example URI	92
217	A.9.3	Resource Type	92
218	A.9.4	Swagger2.0 Definition	92
219	A.9.5	Property Definition	95
220	A.9.6	CRUDN behavior	95
221	A.10	Heating Zone Mapping	95
222	A.10.1	Introduction	95
223	A.10.2	Example URI	96
224	A.10.3	Resource Type	96
225	A.10.4	Swagger2.0 Definition	96
226	A.10.5	Property Definition	97
227	A.10.6	CRUDN behavior	98
228	A.11	HVAC Fan Mode Mapping	98
229	A.11.1	Introduction	98
230	A.11.2	Example URI	98
231	A.11.3	Resource Type	98
232	A.11.4	Swagger2.0 Definition	98
233	A.11.5	Property Definition	101
234	A.11.6	CRUDN behavior	101
235	A.12	On Off Mapping	101
236	A.12.1	Introduction	101
237	A.12.2	Example URI	101
238	A.12.3	Resource Type	101

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

265
266

No table of figures entries found.

Figures

267

268

269

270

271

272

273

Tables

Table 6-1 AllJoyn to OCF Device Type Mapping..... 13

Table 7-1 AllJoyn Interface to OCF Resource Type Mapping – Minimum Interface Set 16

Table 7-2 AllJoyn Interface to OCF Resource Type Mapping – Optional Interface Set 17

Table 8-1 Interface to Resource Summary..... 18

1 Scope

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

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

2 Normative references

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

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

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

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

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

https://www.iab.org/wp-content/IAB-uploads/2016/03/OCF-Derived-Models-for-Interoperability-Between-IoT-Ecosystems_v2-examples.pdf

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

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

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

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

3 Terms, definitions symbols and abbreviations

3.1 Terms and definitions

3.2 Symbols and abbreviations

3.2.1

OCF

Open Connectivity Foundation

The organization that created these specifications

3.2.2

RAML

RESTful API Modelling Language

RAML is a simple and succinct way of describing practically-RESTful APIs. See RAML.

3.3 Conventions

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

4 Document conventions and organization

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

4.1 Notation

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

Required (or shall or mandatory).

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

Recommended (or should).

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

Allowed (or allowed).

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

Conditionally allowed (CA)

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

Conditionally required (CR)

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

DEPRECATED

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

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

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

Words that are emphasized are printed in *italic*.

4.2 Data types

See OCF Core Specification.

5 Theory of Operation

5.1 Interworking Approach

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

5.2 Mapping Syntax

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

5.2.1 General

All statements are terminated with a carriage return.

5.2.2 Value Assignment

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

5.2.3 Property Naming

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

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

5.2.4 Arrays

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

5.2.5 Default Mapping

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

5.2.6 Conditional Mapping

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

398 if 'condition', 'mapping'.

399 Is applied.

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

401 5.2.7 Loops

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

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

405 Where:

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

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

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

409 Is applied.

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

411 modearray[supportedmodes[x]]

412 5.2.8 Method Invocation

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

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

420 6 Device Type Mapping

421 6.1 Introduction

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

423 6.2 AllJoyn Device Types to OCF Device Types

424 The following table captures the equivalency mapping between AllJoyn defined Device Types (see
425 AllJoyn Common Data Model Interface Definitions) and OCF defined Device Types (see Table 10-
426 1 in OCF Smart Home Device Specification). The minimum interface set for the AllJoyn definitions
427 is provided in the HAE Theory of Operation; the minimum Resource sets for each OCF Device is
428 provided in OCF Smart Home Device Specification.

429 **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

430

431 6.3 OCF Device Types with no AllJoyn Equivalent

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

Table 6-2 OCF Device Types with no AllJoyn	OCF Device Type
--	-----------------

EquivalentOCF Device Name	
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

434

435 **7 Resource to Interface Equivalence**

436 **7.1 Introduction**

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

439 **7.2 AllJoyn Interfaces to OCF Resources**

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

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

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

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

449

450

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

8 Detailed Mapping APIs

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

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

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

Table 8-1 Interface to Resource Summary

AllJoyn Interface Name	Equivalent Resource(s)	Mapping Section
Environment.CurrentAirQuality	oic.r.airqualitycollection	8.1
Environment.CurrentAirQualityLevel	oic.r.airqualitycollection	8.2
Environment.CurrentHumidity	oic.r.humidity	8.3
Environment.CurrentTemperature	oic.r.temperature	8.4
Environment.TargetHumidity	oic.r.humidity, oic.r.selectablelevels	8.5

Environment.TargetTemperature	oic.r.temperature	8.6
Operation.AudioVolume	oic.r.audio	8.7
Operation.ClimateControlMode	oic.r.mode, oic.r.operationalstate	8.8
Operation.ClosedStatus	oic.r.door	8.9
Operation.CycleControl	oic.r.operational.state	8.10
Operation.FanSpeedLevel	oic.r.airflow	8.11
Operation.HeatingZone	oic.r.heatingzonecollection	8.12
Operation.HvacFanMode	oic.r.mode	8.13
Operation.OnOffStatus, Operation.OnControl, Operation.OffControl	oic.r.switch.binary	8.14
Operation.OvenCyclePhase	oic.r.operationalstate	8.15

463

464 **8.1 Air Quality Mapping**

465 **8.1.1 Introduction**

466 This API defines the mapping between the AllJoyn AirQuality interface and the OCF AirQuality
467 Resource. If more than one instance of the AirQuality interface is exposed then each instance
468 maps to an instance of the OCF AirQuality Resource. The mapping defined in the schema
469 describes the population of the OCF AirQuality Resource. Even if there is only a single instance
470 of an OCF AirQuality Resource this shall be included in an instance of an OCF AirQualityCollection.
471 The number of links in the collection equates to the number of instances of the AllJoyn
472 CurrentAirQuality interface that are exposed. When mapping from OCF the valueType of the
473 Resource shall be introspected, this API is invoked only if this is set to 'Measured'

474 **8.1.2 Example URI**

475 /CurrentAirQualityResURI

476 **8.1.3 Resource Type**

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

478 **8.1.4 RAML Definition**

```
479 #%RAML 0.8
480 title: CurrentAirQualityInterfaceMapping
481 version: OCFv1.0.0-20170317
482 traits:
483   - interface-sensor :
484     queryParameters:
485       if:
486         enum: ["oic.if.s", "oic.if.baseline"]
```

487

```

488 /CurrentAirQualityResURI:
489     description: |
490         This API defines the mapping between the AllJoyn AirQuality interface and the OCF AirQuality
491         Resource.
492         If more than one instance of the AirQuality interface is exposed then each instance maps to an
493         instance of the OCF AirQuality Resource.
494         The mapping defined in the schema describes the population of the OCF AirQuality Resource.
495         Even if there is only a single instance of an OCF AirQuality Resource this shall be included in
496         an instance of an OCF AirQualityCollection.
497         The number of links in the collection equates to the number of instances of the AllJoyn
498         CurrentAirQuality interface that are exposed.
499         When mapping from OCF the valueType of the Resource shall be introspected, this API is invoked
500         only if this is set to 'Measured'
501
502     is : ['interface-sensor']
503
504     get:
505         responses :
506             200:
507                 body:
508                     application/json:
509                         schema: |
510                             {
511 "http://openinterconnect.org/asamapping/schemas/asa.environment.currentairquality.json#",
512 "$schema": "http://json-schema.org/draft-04/schema#",
513 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
514 reserved.",
515 "title": "Current Air Quality",
516 "definitions": {
517     "asa.environment.currentairquality": {
518         "type": "object",
519         "properties": {
520             "contaminanttype": {
521                 "type": "integer",
522                 "description": "The contaminant type",
523                 "x-ocf-conversion": {
524                     "x-ocf-alias": "oic.r.airquality",
525                     "x-to-ocf": [
526                         "valuetype = Measured",
527                         "contaminanttypearray = [CH2O,CO2,CO,PM2_5,PM10,VOC]",
528                         "ocf.contaminanttype = contaminanttypearray[contaminanttype]"
529                     ],
530                     "x-from-ocf": [
531                         "contaminanttype = indexof contaminanttypearray[ocf.contaminanttype]"
532                     ]
533                 }
534             },
535             "currentvalue":{
536                 "type": "number",
537                 "x-ocf-conversion": {
538                     "x-ocf-alias": "oic.r.airquality",
539                     "x-to-ocf": [
540                         "contaminantvalue = currentvalue"
541                     ],
542                     "x-from-ocf": [
543                         "currentvalue = contaminantvalue"
544                     ]
545                 }
546             },
547             "minvalue": {
548                 "type": "number",
549                 "x-ocf-conversion": {
550                     "x-ocf-alias": "oic.r.airquality",
551                     "x-to-ocf": [
552                         "range[0] = minvalue"
553                     ],
554                     "x-from-ocf": [

```

```

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

```

8.1.5 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,VOC]ocf.contaminanttype = contaminanttypearray[contaminanttype]	contaminanttype = indexof contaminanttypearray[ocf.contaminanttype]	The contaminant type

8.1.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentAirQualityResURI		get			

8.2 Air Quality Level Mapping

8.2.1 Introduction

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

8.2.2 Example URI

/CurrentAirQualityLevelResURI

8.2.3 Resource Type

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

8.2.4 RAML Definition

```

#%RAML 0.8
title: CurrentAirQualityLevelInterfaceMapping
version: OCFv1.0.0-20170317

traits:
  - interface-sensor :
      queryParameters:
        if:
          enum: ["oic.if.s", "oic.if.baseline"]

/CurrentAirQualityLevelResURI:
  description: |
    This API defines the mapping between the AllJoyn AirQualityLevel interface and the OCF AirQuality Resource.
    If more than one instance of the AirQualityLevel interface is exposed then each instance maps to an instance of the OCF AirQuality Resource.
    The mapping defined in the schema describes the population of the OCF AirQuality Resource.
    Even if there is only a single instance of an OCF AirQuality Resource then this shall be included in an instance of an OCF AirQualityCollection.
    The number of links in the collection equates to the number of instances of the AllJoyn CurrentAirQuality interface that are exposed.
    When mapping from OCF the valueType of the Resource shall be introspected, this API is invoked

```

```

649 only if this is set to 'Qualitative'
650
651 is : ['interface-sensor']
652
653 get:
654     responses :
655         200:
656             body:
657                 application/json:
658                     schema: |
659                         {
660                             "id":
661 "http://openinterconnect.org/asamapping/schemas/asa.environment.currentairqualitylevel.json#",
662                             "$schema": "http://json-schema.org/draft-04/schema#",
663                             "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
664 reserved.",
665                             "title": "Current Air Quality Level",
666                             "definitions": {
667                                 "asa.environment.currentairqualitylevel": {
668                                     "type": "object",
669                                     "properties": {
670                                         "contaminanttype": {
671                                             "type": "integer",
672                                             "description": "The contaminant type",
673                                             "x-ocf-conversion": {
674                                                 "x-ocf-alias": "oic.r.airquality",
675                                                 "x-to-ocf": [
676                                                     "valuetype = Qualitative",
677                                                     "if contaminanttype = 0, ocf.contaminanttype = CH2O",
678                                                     "if contaminanttype = 1, ocf.contaminanttype = CO2",
679                                                     "if contaminanttype = 2, ocf.contaminanttype = CO",
680                                                     "if contaminanttype = 3, ocf.contaminanttype = PM2_5",
681                                                     "if contaminanttype = 4, ocf.contaminanttype = PM10",
682                                                     "if contaminanttype = 5, ocf.contaminanttype = VOC",
683                                                     "if contaminanttype = 253, ocf.contaminanttype = Smoke",
684                                                     "if contaminanttype = 254, ocf.contaminanttype = Odor",
685                                                     "if contaminanttype = 255, ocf.contaminanttype = AirPollution"
686                                                 ],
687                                                 "x-from-ocf": [
688                                                     "if ocf.contaminanttype = CH2O, contaminanttype = 0",
689                                                     "if ocf.contaminanttype = CO2, contaminanttype = 1",
690                                                     "if ocf.contaminanttype = CO, contaminanttype = 2",
691                                                     "if ocf.contaminanttype = PM2_5, contaminanttype = 3",
692                                                     "if ocf.contaminanttype = PM10, contaminanttype = 4",
693                                                     "if ocf.contaminanttype = VOC, contaminanttype = 5",
694                                                     "if ocf.contaminanttype = Smoke, contaminanttype = 253",
695                                                     "if ocf.contaminanttype = Odor, contaminanttype = 254",
696                                                     "if ocf.contaminanttype = AirPollution, contaminanttype = 255"
697                                                 ]
698                                             },
699                                         "currentlevel": {
700                                             "type": "integer",
701                                             "x-ocf-conversion": {
702                                                 "x-ocf-alias": "oic.r.airquality",
703                                                 "x-to-ocf": [
704                                                     "contaminantvalue = currentlevel"
705                                                 ],
706                                                 "x-from-ocf": [
707                                                     "currentlevel = contaminantvalue"
708                                                 ]
709                                             }
710                                         },
711                                         "maxlevel": {
712                                             "type": "integer",
713                                             "x-ocf-conversion": {
714                                                 "x-ocf-alias": "oic.r.airquality",
715                                                 "x-to-ocf": [
716                                                     "range[0] = 0",

```



```

717         "range[1] = maxvalue"
718     ],
719     "x-from-ocf": [
720         "maxvalue = range[1]"
721     ]
722 }
723 }
724 }
725 }
726 },
727 "type": "object",
728 "allOf": [
729     {"$ref": "#/definitions/asa.environment.currentairqualitylevel"}
730 ],
731 "required": ["contaminanttype", "currentlevel", "maxlevel"]
732 }
733
734 example: |
735 {
736     "rt":      ["oic.r.airqualitycollection"]
737 }
738

```

8.2.5 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	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	The contaminant type

		contaminanttype = 254, ocf.contaminanttype = Odorif contaminanttype = 255, ocf.contaminanttype = AirPollution	= Odor, contaminanttype = 254if ocf.contaminanttype = AirPollution, contaminanttype = 255	
--	--	--	--	--

8.2.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentAirQualityLevelResURI		get			

8.3 Current Humidity Mapping

8.3.1 Introduction

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

8.3.2 Example URI

/CurrentHumidityResURI

8.3.3 Resource Type

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

8.3.4 RAML Definition

```

#%RAML 0.8

title: CurrentHumidityInterfaceMapping
version: OCFv1.0.0-20170317

traits:
- interface-sensor :
    queryParameters:
        if:
            enum: ["oic.if.s", "oic.if.baseline"]

/CurrentHumidityResURI:

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

    is : ['interface-sensor']

    get:
        responses :
            200:
                body:
                    application/json:
                        schema: |
                            {
                                "id":
"111http://openinterconnect.org/asamapping/schemas/asa.environment.currenthumidity.json#",
112    "$schema": "http://json-schema.org/draft-04/schema#",
113    "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
114    reserved.",
115    "title": "Current Humidity",

```

```

822     "definitions": {
823       "asa.environment.currenthumidity": {
824         "type": "object",
825         "properties": {
826           "currentvalue": {
827             "type": "number",
828             "description": "Measured value",
829             "x-ocf-conversion": {
830               "x-ocf-alias": "oic.r.humidity",
831               "x-to-ocf": [
832                 "humidity = currentValue"
833               ],
834               "x-from-ocf": [
835                 "currentvalue = humidity"
836               ]
837             }
838           },
839           "maxvalue": {
840             "type": "number",
841             "description": "Max measured value for humidity",
842             "x-ocf-conversion": {
843               "x-ocf-alias": "oic.r.humidity",
844               "x-to-ocf": [
845                 "range[0] = 0",
846                 "range[1] = maxvalue"
847               ],
848               "x-from-ocf": [
849                 "maxvalue = range[1]"
850               ]
851             }
852           }
853         }
854       },
855       "type": "object",
856       "allOf": [
857         {"$ref": "#/definitions/asa.environment.currenthumidity"}
858       ],
859       "required": [ "currentvalue", "maxvalue" ]
860     }
861
862     example: |
863       {
864         "rt":      ["oic.r.humidity"]
865       }

```

8.3.5 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] range[1] maxvalue	= maxvalue range[1] =	Max measured value for humidity

8.3.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentHumidityResURI		get			

8.4 Current Temperature Mapping

8.4.1 Introduction

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

8.4.2 Example URI

/CurrentTemperatureResURI

8.4.3 Resource Type

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

8.4.4 RAML Definition

```
##RAML 0.8

title: CurrentTemperatureInterfaceMapping
version: OCFv1.0.0-20170317

traits:
  - interface-sensor :
      queryParameters:
        if:
          enum: ["oic.if.s", "oic.if.baseline"]

/CurrentTemperatureResURI:

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

  is : ['interface-sensor']

  get:
    responses :
      200:
        body:
          application/json:
            schema: |
              {
                "id":
                  "http://openinterconnect.org/asamapping/schemas/asa.environment.currenttemperature.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Current Temperature",
                "definitions": {
                  "asa.environment.currenttemperature": {
                    "type": "object",
                    "properties": {
                      "currentvalue": {
                        "type": "number",
                        "description": "Measured value",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.temperature",
                          "x-to-ocf": [
                            "temperature = currentValue",
                            "units = C"
                          ],
                          "x-from-ocf": {
                            "oneOf": [
                              {
```

```

887         "properties": {
888             "units": "string",
889             "enum": ["C"]
890         },
891         "x-from-ocf": [
892             "currentvalue = temperature"
893         ]
894     },
895     {
896         "properties": {
897             "units": "string",
898             "enum": ["F"]
899         },
900         "x-from-ocf": [
901             "currentvalue = (temperature-32)*5/9"
902         ]
903     },
904     {
905         "properties": {
906             "units": "string",
907             "enum": ["K"]
908         },
909         "x-from-ocf": [
910             "currentvalue = temperature-273.15"
911         ]
912     }
913 ]
914 }
915 }
916 },
917 "precision": {
918     "type": "number",
919     "x-ocf-conversion": {
920         "x-ocf-alias": "oic.r.temperature",
921         "x-to-ocf": [
922             "ocf.precision = precision"
923         ],
924         "x-from-ocf": [
925             "precision = ocf.precision"
926         ]
927     }
928 },
929 "updatemintime": {
930     "type": "integer",
931     "x-ocf-conversion": {
932         "x-ocf-alias": "oic.r.value.conditional",
933         "x-to-ocf": [
934             "ocf.minnotifyperiod = updatemintime"
935         ],
936         "x-from-ocf": [
937             "updatemintime = ocf.minnotifyperiod"
938         ]
939     }
940 }
941 }
942 }
943 },
944 "type": "object",
945 "allOf": [
946     {"$ref": "#/definitions/asa.environment.currenttemperature"}
947 ],
948 "required": [ "currentvalue", "precision", "updatemintime" ]
949 }
950
951 example: |
952 {
953     "rt":      ["oic.r.temperature"]
954 }
955

```

8.4.5 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	

8.4.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentTemperatureResURI		get			

8.5 Target Humidity Mapping

8.5.1 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.

8.5.2 Example URI

/TargetHumidityResURI

8.5.3 Resource Type

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

8.5.4 RAML Definition

```
##RAML 0.8
title: TargetHumidityInterfaceMapping
version: OCFv1.0.0-20170317

traits:
  - interface-actuator :
      queryParameters:
        if:
          enum: ["oic.if.a", "oic.if.baseline"]

/TargetHumidityResURI:
  description: |
    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.

  is : ['interface-actuator']

  get:
    responses :
      200:
        body:
          application/json:
            schema: |
              {
                "id":
```

```

993 "http://openinterconnect.org/asamapping/schemas/asa.environment.targethumidity.json#",
994   "$schema": "http://json-schema.org/draft-04/schema#",
995   "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
996 reserved.",
997   "title": "Target Humidity",
998   "definitions": {
999     "asa.environment.targethumidity": {
1000       "type": "object",
1001       "properties": {
1002         "targetvalue": {
1003           "type": "number",
1004           "description": "Measured value",
1005           "x-ocf-conversion": {
1006             "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
1007             "x-to-ocf": [
1008               "if minvalue != maxvalue, ocf.desiredhumidity =
1009 targetvalue;ocf.targetlevel = selectablehumiditylevels[0].",
1010               "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
1011             ],
1012             "x-from-ocf": [
1013               "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
1014               "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
1015             ]
1016           }
1017         },
1018         "minvalue": {
1019           "type": "number",
1020           "x-ocf-conversion": {
1021             "x-ocf-alias": "oic.r.humidity",
1022             "x-to-ocf": [
1023               "range[0] = minvalue"
1024             ],
1025             "x-from-ocf": [
1026               "minvalue = range[0]",
1027               "otherwise: minvalue = 0"
1028             ]
1029           }
1030         },
1031         "maxvalue": {
1032           "type": "number",
1033           "x-ocf-conversion": {
1034             "x-ocf-alias": "oic.r.humidity",
1035             "x-to-ocf": [
1036               "range[1] = maxvalue"
1037             ],
1038             "x-from-ocf": [
1039               "maxvalue = range[1]",
1040               "otherwise: maxvalue = 100"
1041             ]
1042           }
1043         },
1044         "stepvalue": {
1045           "type": "number",
1046           "x-ocf-conversion": {
1047             "x-ocf-alias": "oic.r.humidity",
1048             "x-to-ocf": [
1049               "step = stepvalue"
1050             ],
1051             "x-from-ocf": [
1052               "stepvalue = step",
1053               "otherwise: stepvalue = 1"
1054             ]
1055           }
1056         },
1057         "selectablehumiditylevels": {
1058           "type": "array",
1059           "items": {
1060             "type": "number"
1061           },
1062           "x-ocf-conversion": {
1063             "x-ocf-alias": "oic.r.selectablelevels",

```

```

1064         "x-to-ocf": [
1065             "availablelevels[] = selectablehumiditylevels[]"
1066         ],
1067         "x-from-ocf": [
1068             "selectablehumiditylevels[] = availablelevels[]"
1069         ]
1070     }
1071 }
1072 }
1073 }
1074 },
1075 "type": "object",
1076 "allOf": [
1077     {"$ref": "#/definitions/asa.environment.targethumidity"}
1078 ],
1079 "required":
1080 [ "targetvalue", "minvalue", "maxvalue", "stepvalue", "selectablehumiditylevels" ]
1081 }
1082
1083     example: |
1084         {
1085             "rt":      ["oic.r.humidity", "oic.r.selectablelevels"]
1086         }
1087
1088     post:
1089         body:
1090             application/json:
1091
1092         schema: |
1093             {
1094                 "id":
1095 "http://openinterconnect.org/asamapping/schemas/asa.environment.targethumidity.json#",
1096                 "$schema": "http://json-schema.org/draft-04/schema#",
1097                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1098 reserved.",
1099                 "title": "Target Humidity",
1100                 "definitions": {
1101                     "asa.environment.targethumidity": {
1102                         "type": "object",
1103                         "properties": {
1104                             "targetvalue": {
1105                                 "type": "number",
1106                                 "description": "Measured value",
1107                                 "x-ocf-conversion": {
1108                                     "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
1109                                     "x-to-ocf": [
1110                                         "if minvalue != maxvalue, ocf.desiredhumidity = targetvalue;ocf.targetlevel
1111 = selectablehumiditylevels[0].",
1112                                         "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
1113                                     ],
1114                                     "x-from-ocf": [
1115                                         "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
1116                                         "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
1117                                     ]
1118                                 }
1119                             },
1120                             "minvalue": {
1121                                 "type": "number",
1122                                 "x-ocf-conversion": {
1123                                     "x-ocf-alias": "oic.r.humidity",
1124                                     "x-to-ocf": [
1125                                         "range[0] = minvalue"
1126                                     ],
1127                                     "x-from-ocf": [
1128                                         "minvalue = range[0]",
1129                                         "otherwise: minvalue = 0"
1130                                     ]
1131                                 }
1132                             }
1133                         }
1134                     }
1135                 }

```



```

1132         "maxvalue": {
1133             "type": "number",
1134             "x-ocf-conversion": {
1135                 "x-ocf-alias": "oic.r.humidity",
1136                 "x-to-ocf": [
1137                     "range[1] = maxvalue"
1138                 ],
1139                 "x-from-ocf": [
1140                     "maxvalue = range[1]",
1141                     "otherwise: maxvalue = 100"
1142                 ]
1143             }
1144         },
1145         "stepvalue": {
1146             "type": "number",
1147             "x-ocf-conversion": {
1148                 "x-ocf-alias": "oic.r.humidity",
1149                 "x-to-ocf": [
1150                     "step = stepvalue"
1151                 ],
1152                 "x-from-ocf": [
1153                     "stepvalue = step",
1154                     "otherwise: stepvalue = 1"
1155                 ]
1156             }
1157         },
1158         "selectablehumiditylevels": {
1159             "type": "array",
1160             "items": {
1161                 "type": "number"
1162             },
1163             "x-ocf-conversion": {
1164                 "x-ocf-alias": "oic.r.selectablelevels",
1165                 "x-to-ocf": [
1166                     "availablelevels[] = selectablehumiditylevels[]"
1167                 ],
1168                 "x-from-ocf": [
1169                     "selectablehumiditylevels[] = availablelevels[]"
1170                 ]
1171             }
1172         }
1173     }
1174 },
1175 {
1176     "type": "object",
1177     "allOf": [
1178         {"$ref": "#/definitions/asa.environment.targethumidity"}
1179     ],
1180     "required":
1181 [ "targetvalue", "minvalue", "maxvalue", "stepvalue", "selectablehumiditylevels" ]
1182     }
1183
1184     responses :
1185         200:
1186             body:
1187                 application/json:
1188                     schema: |
1189                         {
1190                             "id":
1191 "http://openinterconnect.org/asamapping/schemas/asa.environment.targethumidity.json#",
1192                             "$schema": "http://json-schema.org/draft-04/schema#",
1193                             "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1194 reserved.",
1195                             "title": "Target Humidity",
1196                             "definitions": {
1197                                 "asa.environment.targethumidity": {
1198                                     "type": "object",
1199                                     "properties": {

```

```

1200         "targetvalue": {
1201             "type": "number",
1202             "description": "Measured value",
1203             "x-ocf-conversion": {
1204                 "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
1205                 "x-to-ocf": [
1206                     "if minvalue != maxvalue, ocf.desiredhumidity =
1207 targetvalue;ocf.targetlevel = selectablehumiditylevels[0].",
1208                     "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
1209                 ],
1210                 "x-from-ocf": [
1211                     "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
1212                     "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
1213                 ]
1214             }
1215         },
1216         "minvalue": {
1217             "type": "number",
1218             "x-ocf-conversion": {
1219                 "x-ocf-alias": "oic.r.humidity",
1220                 "x-to-ocf": [
1221                     "range[0] = minvalue"
1222                 ],
1223                 "x-from-ocf": [
1224                     "minvalue = range[0]",
1225                     "otherwise: minvalue = 0"
1226                 ]
1227             }
1228         },
1229         "maxvalue": {
1230             "type": "number",
1231             "x-ocf-conversion": {
1232                 "x-ocf-alias": "oic.r.humidity",
1233                 "x-to-ocf": [
1234                     "range[1] = maxvalue"
1235                 ],
1236                 "x-from-ocf": [
1237                     "maxvalue = range[1]",
1238                     "otherwise: maxvalue = 100"
1239                 ]
1240             }
1241         },
1242         "stepvalue": {
1243             "type": "number",
1244             "x-ocf-conversion": {
1245                 "x-ocf-alias": "oic.r.humidity",
1246                 "x-to-ocf": [
1247                     "step = stepvalue"
1248                 ],
1249                 "x-from-ocf": [
1250                     "stepvalue = step",
1251                     "otherwise: stepvalue = 1"
1252                 ]
1253             }
1254         },
1255         "selectablehumiditylevels": {
1256             "type": "array",
1257             "items": {
1258                 "type": "number"
1259             },
1260             "x-ocf-conversion": {
1261                 "x-ocf-alias": "oic.r.selectablelevels",
1262                 "x-to-ocf": [
1263                     "availablelevels[] = selectablehumiditylevels[]"
1264                 ],
1265                 "x-from-ocf": [
1266                     "selectablehumiditylevels[] = availablelevels[]"
1267                 ]
1268             }
1269         }
1270     }

```

```

1271     }
1272   },
1273   "type": "object",
1274   "allOf": [
1275     { "$ref": "#/definitions/asa.environment.targethumidity" }
1276   ],
1277   "required":
1278 [ "targetvalue", "minvalue", "maxvalue", "stepvalue", "selectablehumiditylevels" ]
1279   }
1280
1281   example: |
1282     {
1283       "rt":      ["oic.r.humidity", "oic.r.selectablelevels"]
1284     }
1285

```

8.5.5 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	

8.5.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/TargetHumidityResURI		get	post		

8.6 Target Temperature Mapping

8.6.1 Introduction

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

8.6.2 Example URI

/TargetTemperatureResURI

8.6.3 Resource Type

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

8.6.4 RAML Definition

```
##RAML 0.8
title: TargetTemperatureInterfaceMapping
version: OCFv1.0.0-20170317

traits:
  - interface-actuator :
      queryParameters:
        if:
          enum: ["oic.if.a", "oic.if.baseline"]

/TargetTemperatureResURI:
  description: |
    This API defines the mapping between an instance of an OCF Temperature which exposes only a
    sensor interface
    and the AllJoyn Current Temperature interface.
    A RETRIEVE on a Temperature Sensor maps to an action on an instance of an
    Environment.CurrentTemperature Interface.

  is : ['interface-actuator']

  get:
    responses :
      200:
        body:
          application/json:
            schema: |
              {
                "id":
                  "http://openinterconnect.org/asamapping/schemas/asa.environment.targettemperature.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Target Temperature",
                "definitions": {
                  "asa.environment.targettemperature": {
                    "type": "object",
                    "properties": {
                      "targetvalue": {
                        "type": "number",
                        "description": "Measured value",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.temperature",
                          "x-to-ocf": [
                            "temperature = targetvalue",
                            "units = C"
                          ],
                        },
                      "x-from-ocf": {
                        "oneOf": [
                          {
                            "properties": {
                              "units": "string",
                              "enum": ["C"]
                            },
                          },
                          {
                            "x-from-ocf": [
                              "targetvalue = temperature"
                            ]
                          }
                        ],
                      },
                      {
                        "properties": {
                          "units": "string",
                          "enum": ["F"]
                        }
                      }
                    }
                  }
                }
              }
```

```

1357         },
1358         "x-from-ocf": [
1359             "targetvalue = (temperature-32)*5/9"
1360         ],
1361     },
1362     {
1363         "properties": {
1364             "units": "string",
1365             "enum": ["K"]
1366         },
1367         "x-from-ocf": [
1368             "targetvalue = temperature-273.15"
1369         ]
1370     }
1371 ]
1372 }
1373 }
1374 },
1375 "minvalue": {
1376     "type": "number",
1377     "x-ocf-conversion": {
1378         "x-ocf-alias": "oic.r.temperature",
1379         "x-to-ocf": [
1380             "range[0] = minvalue"
1381         ],
1382         "x-from-ocf": [
1383             "minvalue = range[0]",
1384             "otherwise: minvalue = -MAXINT"
1385         ]
1386     }
1387 },
1388 "maxvalue": {
1389     "type": "number",
1390     "x-ocf-conversion": {
1391         "x-ocf-alias": "oic.r.temperature",
1392         "x-to-ocf": [
1393             "range[1] = maxvalue"
1394         ],
1395         "x-from-ocf": [
1396             "maxvalue = range[1]",
1397             "otherwise: maxvalue = MAXINT"
1398         ]
1399     }
1400 },
1401 "step": {
1402     "type": "number",
1403     "x-ocf-conversion": {
1404         "x-ocf-alias": "oic.r.temperature",
1405         "x-to-ocf": [
1406             "ocf.step = step"
1407         ],
1408         "x-from-ocf": [
1409             "step = ocf.step",
1410             "otherwise: step = undefined (0x00)"
1411         ]
1412     }
1413 }
1414 }
1415 }
1416 },
1417 "type": "object",
1418 "allOf": [
1419     {"$ref": "#/definitions/asa.environment.targettemperature"}
1420 ],
1421 "required": [ "targetvalue", "minvalue", "maxvalue", "step" ]
1422 }
1423
1424 example: |
1425 {
1426     "rt":      ["oic.r.temperature"]

```

```

1427         }
1428
1429     post:
1430     body:
1431         application/json:
1432
1433         schema: |
1434             {
1435                 "id":
1436                 "http://openinterconnect.org/asamapping/schemas/asa.environment.targettemperature.json#",
1437                 "$schema": "http://json-schema.org/draft-04/schema#",
1438                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1439 reserved.",
1440                 "title": "Target Temperature",
1441                 "definitions": {
1442                     "asa.environment.targettemperature": {
1443                         "type": "object",
1444                         "properties": {
1445                             "targetvalue": {
1446                                 "type": "number",
1447                                 "description": "Measured value",
1448                                 "x-ocf-conversion": {
1449                                     "x-ocf-alias": "oic.r.temperature",
1450                                     "x-to-ocf": [
1451                                         "temperature = targetvalue",
1452                                         "units = C"
1453                                     ],
1454                                     "x-from-ocf": {
1455                                         "oneOf": [
1456                                             {
1457                                                 "properties": {
1458                                                     "units": "string",
1459                                                     "enum": ["C"]
1460                                                 },
1461                                                 "x-from-ocf": [
1462                                                     "targetvalue = temperature"
1463                                                 ]
1464                                             },
1465                                             {
1466                                                 "properties": {
1467                                                     "units": "string",
1468                                                     "enum": ["F"]
1469                                                 },
1470                                                 "x-from-ocf": [
1471                                                     "targetvalue = (temperature-32)*5/9"
1472                                                 ]
1473                                             },
1474                                             {
1475                                                 "properties": {
1476                                                     "units": "string",
1477                                                     "enum": ["K"]
1478                                                 },
1479                                                 "x-from-ocf": [
1480                                                     "targetvalue = temperature-273.15"
1481                                                 ]
1482                                             }
1483                                         ]
1484                                     }
1485                                 },
1486                                 "minvalue": {
1487                                     "type": "number",
1488                                     "x-ocf-conversion": {
1489                                         "x-ocf-alias": "oic.r.temperature",
1490                                         "x-to-ocf": [
1491                                             "range[0] = minvalue"
1492                                         ],
1493                                         "x-from-ocf": [
1494                                             "minvalue = range[0]",
1495                                             "otherwise: minvalue = -MAXINT"

```

```

1496         ]
1497     }
1498 },
1499     "maxvalue": {
1500         "type": "number",
1501         "x-ocf-conversion": {
1502             "x-ocf-alias": "oic.r.temperature",
1503             "x-to-ocf": [
1504                 "range[1] = maxvalue"
1505             ],
1506             "x-from-ocf": [
1507                 "maxvalue = range[1]",
1508                 "otherwise: maxvalue = MAXINT"
1509             ]
1510         }
1511     },
1512     "step": {
1513         "type": "number",
1514         "x-ocf-conversion": {
1515             "x-ocf-alias": "oic.r.temperature",
1516             "x-to-ocf": [
1517                 "ocf.step = step"
1518             ],
1519             "x-from-ocf": [
1520                 "step = ocf.step",
1521                 "otherwise: step = undefined (0x00)"
1522             ]
1523         }
1524     }
1525 }
1526 },
1527 },
1528 "type": "object",
1529 "allOf": [
1530     {"$ref": "#/definitions/asa.environment.targettemperature"}
1531 ],
1532 "required": [ "targetvalue", "minvalue", "maxvalue", "step" ]
1533 }
1534
1535 responses :
1536 200:
1537     body:
1538         application/json:
1539             schema: |
1540                 {
1541                     "id":
1542 "http://openinterconnect.org/asamapping/schemas/asa.environment.targettemperature.json#",
1543                     "$schema": "http://json-schema.org/draft-04/schema#",
1544                     "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1545 reserved.",
1546                     "title": "Target Temperature",
1547                     "definitions": {
1548                         "asa.environment.targettemperature": {
1549                             "type": "object",
1550                             "properties": {
1551                                 "targetvalue": {
1552                                     "type": "number",
1553                                     "description": "Measured value",
1554                                     "x-ocf-conversion": {
1555                                         "x-ocf-alias": "oic.r.temperature",
1556                                         "x-to-ocf": [
1557                                             "temperature = targetvalue",
1558                                             "units = C"
1559                                         ],
1560                                         "x-from-ocf": {
1561                                             "oneOf": [
1562                                                 {
1563                                                     "properties": {

```

```

1564         "units": "string",
1565         "enum": ["C"]
1566     },
1567     "x-from-ocf": [
1568         "targetvalue = temperature"
1569     ]
1570 },
1571 {
1572     "properties": {
1573         "units": "string",
1574         "enum": ["F"]
1575     },
1576     "x-from-ocf": [
1577         "targetvalue = (temperature-32)*5/9"
1578     ]
1579 },
1580 {
1581     "properties": {
1582         "units": "string",
1583         "enum": ["K"]
1584     },
1585     "x-from-ocf": [
1586         "targetvalue = temperature-273.15"
1587     ]
1588 }
1589 ]
1590 }
1591 }
1592 },
1593 "minvalue": {
1594     "type": "number",
1595     "x-ocf-conversion": {
1596         "x-ocf-alias": "oic.r.temperature",
1597         "x-to-ocf": [
1598             "range[0] = minvalue"
1599         ],
1600         "x-from-ocf": [
1601             "minvalue = range[0]",
1602             "otherwise: minvalue = -MAXINT"
1603         ]
1604     }
1605 },
1606 "maxvalue": {
1607     "type": "number",
1608     "x-ocf-conversion": {
1609         "x-ocf-alias": "oic.r.temperature",
1610         "x-to-ocf": [
1611             "range[1] = maxvalue"
1612         ],
1613         "x-from-ocf": [
1614             "maxvalue = range[1]",
1615             "otherwise: maxvalue = MAXINT"
1616         ]
1617     }
1618 },
1619 "step": {
1620     "type": "number",
1621     "x-ocf-conversion": {
1622         "x-ocf-alias": "oic.r.temperature",
1623         "x-to-ocf": [
1624             "ocf.step = step"
1625         ],
1626         "x-from-ocf": [
1627             "step = ocf.step",
1628             "otherwise: step = undefined (0x00)"
1629         ]
1630     }
1631 }
1632 }
1633 },
1634 },

```



```

1635         "type": "object",
1636         "allOf": [
1637             { "$ref": "#/definitions/asa.environment.targettemperature" }
1638         ],
1639         "required": [ "targetvalue", "minvalue", "maxvalue", "step" ]
1640     }
1641

```

8.6.5 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	

8.6.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/TargetTemperatureResURI		get	post		

8.7 Audio Volume Mapping

8.7.1 Introduction

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

8.7.2 Example URI

/AudioVolumeResURI

8.7.3 Resource Type

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

8.7.4 RAML Definition

```

1653 #%RAML 0.8
1654 title: AudioVolumeInterfaceMapping
1655 version: OCFv1.0.0-20170317
1656 traits:
1657   - interface-all :
1658     queryParameters:
1659       if:
1660         enum: ["oic.if.a", "oic.if.baseline"]
1661
1662 /AudioVolumeResURI:
1663   description: |
1664     This API defines the mapping between an instance of an OCF Audio Controls
1665     and the AllJoyn Audio Volume interface.
1666

```

```

1667     is : ['interface-all']
1668
1669     get:
1670         200:
1671             body:
1672                 application/json:
1673                     schema: |
1674                         {
1675                             "id":
1676 "http://openinterconnect.org/asamapping/schemas/asa.operation.audiovolume.json#",
1677                             "$schema": "http://json-schema.org/draft-04/schema#",
1678                             "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1679 reserved.",
1680                             "title": "Audio Volume",
1681                             "definitions": {
1682                                 "asa.operation.audiovolume": {
1683                                     "type": "object",
1684                                     "properties": {
1685                                         "volume": {
1686                                             "type": "integer",
1687                                             "description": "Speaker volume index",
1688                                             "x-ocf-conversion": {
1689                                                 "x-ocf-alias": "oic.r.audio",
1690                                                 "x-to-ocf": [
1691                                                     "ocf.volume = volume"
1692                                                 ],
1693                                                 "x-from-ocf": [
1694                                                     "volume = ocf.volume"
1695                                                 ]
1696                                             }
1697                                         },
1698                                         "maxvolume": {
1699                                             "type": "integer",
1700                                             "x-ocf-conversion": {
1701                                                 "x-ocf-alias": "oic.r.audio",
1702                                                 "x-to-ocf": [
1703                                                     "range[0] = 0",
1704                                                     "range[1] = maxvolume"
1705                                                 ],
1706                                                 "x-from-ocf": [
1707                                                     "maxvolume = range[1]",
1708                                                     "otherwise: maxvalue = 100"
1709                                                 ]
1710                                             }
1711                                         },
1712                                         "mute": {
1713                                             "type": "boolean",
1714                                             "x-ocf-conversion": {
1715                                                 "x-ocf-alias": "oic.r.audio",
1716                                                 "x-to-ocf": [
1717                                                     "ocf.mute = mute"
1718                                                 ],
1719                                                 "x-from-ocf": [
1720                                                     "mute = ocf.mute"
1721                                                 ]
1722                                             }
1723                                         }
1724                                     }
1725                                 },
1726                                 "type": "object",
1727                                 "allof": [
1728                                     {"$ref": "#/definitions/asa.operation.audiovolume"}
1729                                 ],
1730                                 "required": [ "volume", "maxvolume", "mute" ]
1731                             }
1732
1733
1734         example: |

```

```

1735         {
1736             "rt":      ["oic.r.audio"]
1737         }
1738
1739     post:
1740     body:
1741     application/json:
1742         schema: |
1743             {
1744                 "id": "http://openinterconnect.org/asamapping/schemas/asa.operation.audiovolume.json#",
1745                 "$schema": "http://json-schema.org/draft-04/schema#",
1746                 "description": "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1747 reserved.",
1748                 "title": "Audio Volume",
1749                 "definitions": {
1750                     "asa.operation.audiovolume": {
1751                         "type": "object",
1752                         "properties": {
1753                             "volume": {
1754                                 "type": "integer",
1755                                 "description": "Speaker volume index",
1756                                 "x-ocf-conversion": {
1757                                     "x-ocf-alias": "oic.r.audio",
1758                                     "x-to-ocf": [
1759                                         "ocf.volume = volume"
1760                                     ],
1761                                     "x-from-ocf": [
1762                                         "volume = ocf.volume"
1763                                     ]
1764                                 }
1765                             },
1766                             "maxvolume": {
1767                                 "type": "integer",
1768                                 "x-ocf-conversion": {
1769                                     "x-ocf-alias": "oic.r.audio",
1770                                     "x-to-ocf": [
1771                                         "range[0] = 0",
1772                                         "range[1] = maxvolume"
1773                                     ],
1774                                     "x-from-ocf": [
1775                                         "maxvolume = range[1]",
1776                                         "otherwise: maxvalue = 100"
1777                                     ]
1778                                 }
1779                             },
1780                             "mute": {
1781                                 "type": "boolean",
1782                                 "x-ocf-conversion": {
1783                                     "x-ocf-alias": "oic.r.audio",
1784                                     "x-to-ocf": [
1785                                         "ocf.mute = mute"
1786                                     ],
1787                                     "x-from-ocf": [
1788                                         "mute = ocf.mute"
1789                                     ]
1790                                 }
1791                             }
1792                         }
1793                     },
1794                     "type": "object",
1795                     "allOf": [
1796                         { "$ref": "#/definitions/asa.operation.audiovolume" }
1797                     ],
1798                     "required": [ "volume", "maxvolume", "mute" ]
1799                 }
1800
1801     responses :

```

```

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

```

8.7.5 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] = 0 range[1] = maxvolume	maxvolume = range[1] otherwise: maxvalue = 100	
mute	oic.r.audio	ocf.mute = mute	mute = ocf.mute	

8.7.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/AudioVolumeResURI		get	post		

8.8 Climate Control Mode Mapping

8.8.1 Introduction

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: mode, supportedmodes -> Mode Resource operationalstate -> OperationalState Resource This can be represented in OCF either as two distinct Resource instances or a single instance with two RTs (oic.r.mode, oic.r.operationalstate)

8.8.2 Example URI

/ClimateControlModeResURI

8.8.3 Resource Type

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

8.8.4 RAML Definition

```

#%RAML 0.8

title: ClimateControlModeInterfaceMapping
version: OCFv1.0.0-20170317

traits:
  - interface-actuator :
      queryParameters:
        if:
          enum: ["oic.if.a", "oic.if.baseline"]

/ClimateControlModeResURI:

  description: |
    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:
    mode, supportedmodes -> Mode Resource
    operationalstate -> OperationalState Resource
    This can be represented in OCF either as two distinct Resource instances or a single instance
    with two RTs (oic.r.mode, oic.r.operationalstate)

  is : ['interface-actuator']

  get:
    responses :
      200:
        body:
          application/json:
            schema: |
              {
                "id":
                  "http://openinterconnect.org/asamapping/schemas/asa.operation.climatecontrolmode.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights

```

```

1912 reserved.",
1913 "title": "Climate Control Mode",
1914 "definitions": {
1915     "asa.operation.climatecontrolmode": {
1916         "type": "object",
1917         "properties": {
1918             "mode": {
1919                 "type": "integer",
1920                 "description": "Current mode of device.",
1921                 "x-ocf-conversion": {
1922                     "x-ocf-alias": "oic.r.mode",
1923                     "x-to-ocf": [
1924                         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
1925                         "ocf.mode[0] = modearray[mode]"
1926                     ],
1927                     "x-from-ocf": [
1928                         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
1929                         "mode = indexof modeArray[ocf.mode[0]]"
1930                     ]
1931                 },
1932             },
1933             "supportedmodes": {
1934                 "type": "array",
1935                 "items": {
1936                     "type": "integer"
1937                 },
1938                 "description": "Array of supported modes",
1939                 "x-ocf-conversion": {
1940                     "x-ocf-alias": "oic.r.mode",
1941                     "x-to-ocf": [
1942                         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
1943                         "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
1944 modearray[supportedmodes[x]]"
1945                     ],
1946                     "x-from-ocf": [
1947                         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
1948                         "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
1949 modearray[ocf.supportedmodes[x]]"
1950                     ]
1951                 },
1952             },
1953             "operationalstate": {
1954                 "type": "integer",
1955                 "description": "Current status of device",
1956                 "x-ocf-conversion": {
1957                     "x-ocf-alias": "oic.r.operationalstate",
1958                     "x-to-ocf": [
1959                         "machinestates =
1960 [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
1961                         "currentmachinestate = machinestates[operationalstate]"
1962                     ],
1963                     "x-from-ocf": [
1964                         "statearray =
1965 [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
1966                         "operationalstate = indexof statearray[currentmachinestate[0]]"
1967                     ]
1968                 },
1969             }
1970         }
1971     },
1972 },
1973 "type": "object",
1974 "allof": [
1975     {"$ref": "#/definitions/asa.operation.climatecontrolmode"}
1976 ],
1977 "required": [ "mode","supportedmodes","operationalstate" ]
1978 }
1979
1980 example: |

```

```

1981         {
1982             "rt":      ["oic.r.mode", "oic.operationalstate"]
1983         }
1984
1985     post:
1986     body:
1987     application/json:
1988         schema: |
1989             {
1990                 "id":
1991 "http://openinterconnect.org/asamapping/schemas/asa.operation.climatecontrolmode.json#",
1992                 "$schema": "http://json-schema.org/draft-04/schema#",
1993                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1994 reserved.",
1995                 "title": "Climate Control Mode",
1996                 "definitions": {
1997                     "asa.operation.climatecontrolmode": {
1998                         "type": "object",
1999                         "properties": {
2000                             "mode": {
2001                                 "type": "integer",
2002                                 "description": "Current mode of device.",
2003                                 "x-ocf-conversion": {
2004                                     "x-ocf-alias": "oic.r.mode",
2005                                     "x-to-ocf": [
2006                                         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2007                                         "ocf.mode[0] = modearray[mode]"
2008                                     ],
2009                                     "x-from-ocf": [
2010                                         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2011                                         "mode = indexof modeArray[ocf.mode[0]]"
2012                                     ]
2013                                 }
2014                             },
2015                             "supportedmodes": {
2016                                 "type": "array",
2017                                 "items": {
2018                                     "type": "integer"
2019                                 },
2020                                 "description": "Array of supported modes",
2021                                 "x-ocf-conversion": {
2022                                     "x-ocf-alias": "oic.r.mode",
2023                                     "x-to-ocf": [
2024                                         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2025                                         "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
2026 modearray[supportedmodes[x]]"
2027                                     ],
2028                                     "x-from-ocf": [
2029                                         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2030                                         "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
2031 modearray[ocf.supportedmodes[x]]"
2032                                     ]
2033                                 }
2034                             },
2035                             "operationalstate": {
2036                                 "type": "integer",
2037                                 "description": "Current status of device",
2038                                 "x-ocf-conversion": {
2039                                     "x-ocf-alias": "oic.r.operationalstate",
2040                                     "x-to-ocf": [
2041                                         "machinestates =
2042 [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
2043                                         "currentmachinestate = machinestates[operationalstate]"
2044                                     ],
2045                                     "x-from-ocf": [
2046                                         "statearray =
2047 [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
2048                                         "operationalstate = indexof statearray[currentmachinestate[0]]"
2049                                     ]

```

```

2050         }
2051     }
2052 }
2053 }
2054 },
2055 "type": "object",
2056 "allOf": [
2057     {"$ref": "#/definitions/asa.operation.climatecontrolmode"}
2058 ],
2059 "required": [ "mode", "supportedmodes", "operationalstate" ]
2060 }
2061
2062 responses :
2063     200:
2064         body:
2065             application/json:
2066                 schema: |
2067                     {
2068                         "id":
2069 "http://openinterconnect.org/asamapping/schemas/asa.operation.climatecontrolmode.json#",
2070                         "$schema": "http://json-schema.org/draft-04/schema#",
2071                         "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2072 reserved.",
2073                         "title": "Climate Control Mode",
2074                         "definitions": {
2075                             "asa.operation.climatecontrolmode": {
2076                                 "type": "object",
2077                                 "properties": {
2078                                     "mode": {
2079                                         "type": "integer",
2080                                         "description": "Current mode of device.",
2081                                         "x-ocf-conversion": {
2082                                             "x-ocf-alias": "oic.r.mode",
2083                                             "x-to-ocf": [
2084                                                 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2085                                                 "ocf.mode[0] = modearray[mode]"
2086                                             ],
2087                                             "x-from-ocf": [
2088                                                 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2089                                                 "mode = indexof modeArray[ocf.mode[0]]"
2090                                             ]
2091                                         }
2092                                     },
2093                                     "supportedmodes": {
2094                                         "type": "array",
2095                                         "items": {
2096                                             "type": "integer"
2097                                         },
2098                                         "description": "Array of supported modes",
2099                                         "x-ocf-conversion": {
2100                                             "x-ocf-alias": "oic.r.mode",
2101                                             "x-to-ocf": [
2102                                                 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2103                                                 "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
2104 modearray[supportedmodes[x]]"
2105                                             ],
2106                                             "x-from-ocf": [
2107                                                 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2108                                                 "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
2109 modearray[ocf.supportedmodes[x]]"
2110                                             ]
2111                                         }
2112                                     },
2113                                     "operationalstate": {
2114                                         "type": "integer",
2115                                         "description": "Current status of device",
2116                                         "x-ocf-conversion": {
2117                                             "x-ocf-alias": "oic.r.operationalstate",

```



```

2118         "x-to-ocf": [
2119             "machinestates =
2120 [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
2121             "currentmachinestate = machinestates[operationalstate]"
2122         ],
2123         "x-from-ocf": [
2124             "statearray =
2125 [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
2126             "operationalstate = indexof statearray[currentmachinestate[0]]"
2127         ]
2128     }
2129 }
2130 }
2131 }
2132 },
2133 "type": "object",
2134 "allOf": [
2135     {"$ref": "#/definitions/asa.operation.climatecontrolmode"}
2136 ],
2137 "required": [ "mode","supportedmodes","operationalstate" ]
2138 }
2139

```

2140 8.8.5 Property Definition

['AllJoyn'] 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.

2141 8.8.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/ClimateControlModeResURI		get	post		

2142 8.9 Closed Status Mapping

2143 8.9.1 Introduction

2144 This API defines the mapping between an instance of an AllJoyn ClosedStatus Interface and the
 2145 OCF Door Resource.

2146 8.9.2 Example URI

2147 /ClosedStatusResURI

8.9.3 Resource Type

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

8.9.4 RAML Definition

```
2151 #RAML 0.8
2152 title: ClosedStatusInterfaceMapping
2153 version: OCFv1.0.0-20170317
2154 traits:
2155   - interface-all :
2156     queryParameters:
2157       if:
2158         enum: ["oic.if.s", "oic.if.baseline"]
2159
2160 /ClosedStatusResURI:
2161   description: |
2162     This API defines the mapping between an instance of an AllJoyn ClosedStatus Interface and
2163     the OCF Door Resource.
2164
2165   is : ['interface-all']
2166   get:
2167     responses :
2168       200:
2169         body:
2170           application/json:
2171             schema: |
2172               {
2173                 "id":
2174 "http://openinterconnect.org/asamapping/schemas/asa.operation.closedstatus.json#",
2175                 "$schema": "http://json-schema.org/draft-04/schema#",
2176                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2177 reserved.",
2178                 "title": "Closed Status",
2179                 "definitions": {
2180                   "asa.operation.closedstatus": {
2181                     "type": "object",
2182                     "properties": {
2183                       "isclosed": {
2184                         "type": "boolean",
2185                         "description": "Open/Closed status Indicator",
2186                         "x-ocf-conversion": {
2187                           "x-ocf-alias": "oic.r.door",
2188                           "x-to-ocf": [
2189                             "if isClosed ocf.openState = Closed.",
2190                             "if !isClosed ocf.openState = Open."
2191                           ],
2192                           "x-from-ocf": [
2193                             "isClosed = (openState == Closed)"
2194                           ]
2195                         }
2196                     }
2197                   }
2198                 },
2199                 "type": "object",
2200                 "allOf": [
2201                   { "$ref": "#/definitions/asa.operation.closedstatus" }
2202                 ],
2203                 "required": [ "isclosed" ]
2204               }
2205
2206   example: |
```

```

2208     {
2209         "rt":      ["oic.r.door"]
2210     }
2211

```

2212 8.9.5 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

2213 8.9.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/ClosedStatusResURI		get			

2214 8.10 Cycle Control Mapping

2215 8.10.1 Introduction

2216 This API defines the mapping between an instance of an AllJoyn CycleControl interface and the
 2217 OCF OperationalState Resource. The AllJoyn interface also supports a Method,
 2218 ExecuteOperationalCommand; this is handled in OCF using an instance of oic.r.actuator within an
 2219 oic.r.action collection. Please see Section 8 of the Mapping Specification for specifics.

2220 8.10.2 Example URI

2221 /CycleControlResURI

2222 8.10.3 Resource Type

2223 The resource type (rt) is defined as: oic.r.operationalstate.

2224 8.10.4 RAML Definition

```

2225 #%RAML 0.8
2226 title: CycleControlInterfaceMapping
2227 version: OCFv1.0.0-20170317
2228 traits:
2229   - interface-sensor :
2230       queryParameters:
2231           if:
2232               enum: ["oic.if.s", "oic.if.baseline"]
2233
2234 /CycleControlResURI:
2235     description: |
2236         This API defines the mapping between an instance of an AllJoyn CycleControl interface and the
2237         OCF OperationalState Resource.
2238         The AllJoyn interface also supports a Method, ExecuteOperationalCommand; this is handled in OCF
2239         using an instance of oic.r.actuator within an oic.r.action collection.
2240         Please see Section 8 of the Mapping Specification for specifics.
2241
2242     is : ['interface-sensor']
2243     get:
2244         responses :
2245             200:
2246                 body:
2247                     application/json:
2248                         schema: |

```

```

2249     {
2250         "id":
2251         "http://openinterconnect.org/asamapping/schemas/asa.operation.ovencyclephase.json#",
2252         "$schema": "http://json-schema.org/draft-04/schema#",
2253         "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2254 reserved.",
2255         "title": "Oven Cycle Phase",
2256         "definitions": {
2257             "asa.operation.ovencyclephase": {
2258                 "type": "object",
2259                 "properties": {
2260                     "cyclephase": {
2261                         "type": "integer",
2262                         "description": "Current phase of the operational cycle",
2263                         "x-ocf-conversion": {
2264                             "x-ocf-alias": "oic.r.operationalstate",
2265                             "x-to-ocf": [
2266                                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
2267                                 "currentmachinestate = phasearray[cyclephase]"
2268                             ],
2269                             "x-from-ocf": [
2270                                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
2271                                 "cyclephase = indexof statearray[currentmachinestate[0]]"
2272                             ]
2273                         }
2274                     },
2275                     "supportedcyclephases": {
2276                         "type": "array",
2277                         "items": {
2278                             "type": "integer"
2279                         },
2280                         "description": "Array of cycle phases supported by the Appliance.",
2281                         "x-ocf-conversion": {
2282                             "x-ocf-alias": "oic.r.operationalstate",
2283                             "x-to-ocf": [
2284                                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
2285                                 "for x=0, x < sizeof(supportedcyclephases): machinestates[x] =
2286 phasearray[supportedcyclephases[x]]"
2287                             ],
2288                             "x-from-ocf": [
2289                                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
2290                                 "for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof
2291 phasearray[machinestates[x]]"
2292                             ]
2293                         }
2294                     },
2295                     "getvendorphasesdescription": {
2296                         "x-ocf-type": "method",
2297                         "description": "Get cycle phases description",
2298                         "x-ocf-conversion": {
2299                             "x-ocf-alias": "oic.r.action"
2300                         }
2301                     }
2302                 }
2303             }
2304         },
2305         "type": "object",
2306         "allOf": [
2307             {"$ref": "#/definitions/asa.operation.ovencyclephase"}
2308         ],
2309         "required": [ "cyclephase","supportedcyclephases" ]
2310     }
2311
2312     example: |
2313     {
2314         "rt":      ["oic.r.operationalstate"]
2315     }
2316

```

2317 **8.10.5 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

2318 **8.10.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/CycleControlResURI		get			

2319 **8.11 Fan Speed Level Mapping**

2320 **8.11.1 Introduction**

2321 This API defines the mapping between an instance of an AllJoyn FanSpeedLevel interface and an
2322 OCF AirFlow Resource. Note that the setting of the FanSpeedLevel to '0x00' (off) is handled via
2323 the 'OffControl' interface rather than writing directly to this interface. In such a case an instance of
2324 Binary Switch shall be exposed on the OCF side; this can be modeled as AirFlowControl which is
2325 then a collection of Binary Switch and AirFlow.

2326 **8.11.2 Example URI**

2327 /FanSpeedLevelResURI

2328 **8.11.3 Resource Type**

2329 The resource type (rt) is defined as: oic.r.airflow.

2330 **8.11.4 RAML Definition**

2331 `##RAML 0.8`
2332 `title: FanSpeedLevelInterfaceMapping`
2333 `version: OCFv1.0.0-20170317`
2334 `traits:`
2335 `- interface-actuator :`
2336 `queryParameters:`

```

2337         if:
2338             enum: ["oic.if.a", "oic.if.baseline"]
2339
2340 /FanSpeedLevelResURI:
2341     description: |
2342         This API defines the mapping between an instance of an AllJoyn FanSpeedLevel interface and an
2343         OCF AirFlow Resource.
2344         Note that the setting of the FanSpeedLevel to '0x00' (off) is handled via the 'OffControl'
2345         interface rather than writing directly to this interface.
2346         In such a case an instance of Binary Switch shall be exposed on the OCF side; this can be
2347         modeled as AirFlowControl which is then a collection of Binary Switch and AirFlow.
2348
2349     is : ['interface-actuator']
2350
2351     get:
2352         responses :
2353             200:
2354                 body:
2355                     application/json:
2356                         schema: |
2357                             {
2358                                 "id":
2359 "http://openinterconnect.org/asamapping/schemas/asa.operation.fanspeedlevel.json#",
2360                                 "$schema": "http://json-schema.org/draft-04/schema#",
2361                                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2362 reserved.",
2363                                 "title": "Fan Speed Level",
2364                                 "definitions": {
2365                                     "asa.operation.fanspeedlevel": {
2366                                         "type": "object",
2367                                         "properties": {
2368                                             "fanspeedlevel": {
2369                                                 "type": "integer",
2370                                                 "description": "Fan speed level. 0 = off.",
2371                                                 "x-ocf-conversion": {
2372                                                     "x-ocf-alias": "oic.r.airflow",
2373                                                     "x-to-ocf": [
2374                                                         "speed = fanspeedlevel"
2375                                                     ],
2376                                                     "x-from-ocf": [
2377                                                         "fanspeedlevel = speed"
2378                                                     ]
2379                                                 },
2380                                             "maxfanspeedlevel": {
2381                                                 "type": "integer",
2382                                                 "description": "Max level allowed for fan speed",
2383                                                 "x-ocf-conversion": {
2384                                                     "x-ocf-alias": "oic.r.airflow",
2385                                                     "x-to-ocf": [
2386                                                         "range[0] = 0",
2387                                                         "range[1] = maxfanspeedlevel"
2388                                                     ],
2389                                                     "x-from-ocf": [
2390                                                         "maxfanspeedlevel = range[1]",
2391                                                         "otherwise: maxfanspeedlevel = 100"
2392                                                     ]
2393                                                 },
2394                                             "automode": {
2395                                                 "type": "integer",
2396                                                 "description": "Auto mode status.",
2397                                                 "x-ocf-conversion": {
2398                                                     "x-ocf-alias": "oic.r.airflow",
2399                                                     "x-to-ocf": [
2400                                                         "if automode != NotSupported(0xFF)",
2401                                                         " ocf.automode = automode",

```

```

2403         "else no mapping"
2404     ],
2405     "x-from-ocf": [
2406         "automode = ocf.automode",
2407         "otherwise: automode = NotSupported(0xFF)"
2408     ]
2409 }
2410 }
2411 }
2412 }
2413 },
2414 "type": "object",
2415 "allOf": [
2416     {"$ref": "#/definitions/asa.operation.fanspeedlevel"}
2417 ],
2418 "required": [ "fanspeedlevel", "maxfanspeedlevel", "automode" ]
2419 }
2420
2421 example: |
2422 {
2423     "rt":      ["oic.r.airflow"]
2424 }
2425
2426 post:
2427     body:
2428         application/json:
2429             schema: |
2430                 {
2431                     "id":
2432 "http://openinterconnect.org/asamapping/schemas/asa.operation.fanspeedlevel.json#",
2433                     "$schema": "http://json-schema.org/draft-04/schema#",
2434                     "description": "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2435 reserved.",
2436                     "title": "Fan Speed Level",
2437                     "definitions": {
2438                         "asa.operation.fanspeedlevel": {
2439                             "type": "object",
2440                             "properties": {
2441                                 "fanspeedlevel": {
2442                                     "type": "integer",
2443                                     "description": "Fan speed level. 0 = off.",
2444                                     "x-ocf-conversion": {
2445                                         "x-ocf-alias": "oic.r.airflow",
2446                                         "x-to-ocf": [
2447                                             "speed = fanspeedlevel"
2448                                         ],
2449                                         "x-from-ocf": [
2450                                             "fanspeedlevel = speed"
2451                                         ]
2452                                     }
2453                                 },
2454                                 "maxfanspeedlevel": {
2455                                     "type": "integer",
2456                                     "description": "Max level allowed for fan speed",
2457                                     "x-ocf-conversion": {
2458                                         "x-ocf-alias": "oic.r.airflow",
2459                                         "x-to-ocf": [
2460                                             "range[0] = 0",
2461                                             "range[1] = maxfanspeedlevel"
2462                                         ],
2463                                         "x-from-ocf": [
2464                                             "maxfanspeedlevel = range[1]",
2465                                             "otherwise: maxfanspeedlevel = 100"
2466                                         ]
2467                                     }
2468                                 },
2469                                 "automode": {
2470                                     "type": "integer",

```

```

2471         "description": "Auto mode status.",
2472         "x-ocf-conversion": {
2473             "x-ocf-alias": "oic.r.airflow",
2474             "x-to-ocf": [
2475                 "if automode != NotSupported(0xFF)",
2476                 " ocf.automode = automode",
2477                 "else no mapping"
2478             ],
2479             "x-from-ocf": [
2480                 "automode = ocf.automode",
2481                 "otherwise: automode = NotSupported(0xFF)"
2482             ]
2483         }
2484     }
2485 }
2486 }
2487 },
2488 "type": "object",
2489 "allOf": [
2490     {"$ref": "#/definitions/asa.operation.fanspeedlevel"}
2491 ],
2492 "required": [ "fanspeedlevel", "maxfanspeedlevel", "automode" ]
2493 }
2494
2495 responses :
2496 200:
2497     body:
2498         application/json:
2499             schema: |
2500                 {
2501                     "id":
2502 "http://openinterconnect.org/asamapping/schemas/asa.operation.fanspeedlevel.json#",
2503                     "$schema": "http://json-schema.org/draft-04/schema#",
2504                     "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2505 reserved.",
2506                     "title": "Fan Speed Level",
2507                     "definitions": {
2508                         "asa.operation.fanspeedlevel": {
2509                             "type": "object",
2510                             "properties": {
2511                                 "fanspeedlevel": {
2512                                     "type": "integer",
2513                                     "description": "Fan speed level. 0 = off.",
2514                                     "x-ocf-conversion": {
2515                                         "x-ocf-alias": "oic.r.airflow",
2516                                         "x-to-ocf": [
2517                                             "speed = fanspeedlevel"
2518                                         ],
2519                                         "x-from-ocf": [
2520                                             "fanspeedlevel = speed"
2521                                         ]
2522                                     }
2523                                 },
2524                                 "maxfanspeedlevel": {
2525                                     "type": "integer",
2526                                     "description": "Max level allowed for fan speed",
2527                                     "x-ocf-conversion": {
2528                                         "x-ocf-alias": "oic.r.airflow",
2529                                         "x-to-ocf": [
2530                                             "range[0] = 0",
2531                                             "range[1] = maxfanspeedlevel"
2532                                         ],
2533                                         "x-from-ocf": [
2534                                             "maxfanspeedlevel = range[1]",
2535                                             "otherwise: maxfanspeedlevel = 100"
2536                                         ]
2537                                     }
2538                                 }
2539                             }
2540                         }
2541                     }
2542                 }

```



```

2539         "automode": {
2540             "type": "integer",
2541             "description": "Auto mode status.",
2542             "x-ocf-conversion": {
2543                 "x-ocf-alias": "oic.r.airflow",
2544                 "x-to-ocf": [
2545                     "if automode != NotSupported(0xFF)",
2546                     " ocf.automode = automode",
2547                     "else no mapping"
2548                 ],
2549                 "x-from-ocf": [
2550                     "automode = ocf.automode",
2551                     "otherwise: automode = NotSupported(0xFF)"
2552                 ]
2553             }
2554         }
2555     }
2556 },
2557 {
2558     "type": "object",
2559     "allOf": [
2560         {"$ref": "#/definitions/asa.operation.fanspeedlevel"}
2561     ],
2562     "required": [ "fanspeedlevel", "maxfanspeedlevel", "automode" ]
2563 }
2564

```

8.11.5 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.

8.11.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/FanSpeedLevelResURI		get	post		

8.12 Heating Zone Mapping

8.12.1 Introduction

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

8.12.2 Example URI

/HeatingZoneResURI

8.12.3 Resource Type

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

8.12.4 RAML Definition

```
##RAML 0.8
title: HeatingZoneInterfaceMapping
version: OCFv1.0.0-20170317

traits:
  - interface-sensor :
    queryParameters:
      if:
        enum: ["oic.if.s", "oic.if.baseline"]

/HeatingZoneResURI:
  description: |
    This API defines the mapping between an instance of an AllJoyn HeatingZone interface and an OCF
    HeatingZoneCollection Resource.
    Each element in the array of heating zones within the AllJoyn HeatingZone interface maps to an
    instance of OCF HeatingZone, itself a link in an instance of an OCF HeatingZoneCollection.
    The mapping defined in the schema describes the population of the OCF HeatingZone Resource that
    constitutes the Resources that are contained in the collection.

  is : ['interface-sensor']

  get:
    responses :
      200:
        body:
          application/json:
            schema: |
              {
                "id":
                  "http://openinterconnect.org/asamapping/schemas/asa.operation.heatingzone.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Heating Zone",
                "definitions": {
                  "asa.operation.heatingzone": {
                    "type": "object",
                    "properties": {
                      "numberOfheatingzones": {
                        "type": "integer",
                        "description": "Number of heating zones.",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.heatingzonecollection",
                          "x-to-ocf": [
                            "number of links in the collection = numberOfheatingzones"
                          ],
                          "x-from-ocf": [
                            "numberOfheatingzones = number of links in the collection"
                          ]
                        }
                      }
                    },
                    "maxheatinglevels": {
                      "type": "array",
                      "items": {
                        "type": "integer"
                      },
                      "description": "Max heating levels for each zone",
                      "x-ocf-conversion": {
                        "x-ocf-alias": "oic.r.heatingzone",
                        "x-to-ocf": [
                          "Instance of oic.r.heatingzone per array item ",
                          "for x=0, x<sizeof(maxheatinglevels): ocf.maxheatinglevel =
maxheatinglevels[x]"
                        ],

```

```

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

```

8.12.5 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(heatingl evels): ocf.heatinglevel = maxheatinglevels [x]	for x=0;x<numlinks(oic.r.heatin gzonecollection): heatinglevels[x] = ocf.heatinglevel	Current heating levels for each zone.
numberofheati ngzones	oic.r.heatingzon ecollection	number of links in the collection = numberofheating zones	numberofheatingzones = number of links in the collection	Number of heating zones.
maxheatinglev els	oic.r.heatingzon e	Instance of oic.r.heatingzone per array item for x=0, x<sizeof(maxheat	for x=0;x<numlinks(oic.r.heatin gzonecollection): maxheatinglevels[x] = ocf.maxheatinglevel	Max heating levels for

		inglevels): ocf.maxheatinglevel = maxheatinglevels [x]		each zone
--	--	---	--	--------------

2683 8.12.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/HeatingZoneResURI		get			

2684 8.13 HVAC Fan Mode Mapping

2685 8.13.1 Introduction

2686 This API defines the mapping between an instance of an AllJoyn HvacFanMode interface and an
2687 OCF Mode Resource.

2688 8.13.2 Example URI

2689 /HvacFanModeResURI

2690 8.13.3 Resource Type

2691 The resource type (rt) is defined as: oic.r.mode.

2692 8.13.4 RAML Definition

```

2693 #%RAML 0.8
2694 title: HvacFanModeInterfaceMapping
2695 version: OCFv1.0.0-20170317
2696 traits:
2697   - interface-actuator :
2698     queryParameters:
2699       if:
2700         enum: ["oic.if.a", "oic.if.baseline"]
2701
2702 /HvacFanModeResURI:
2703   description: |
2704     This API defines the mapping between an instance of an AllJoyn HvacFanMode interface and an OCF
2705     Mode Resource.
2706
2707   is : ['interface-actuator']
2708   get:
2709     responses :
2710       200:
2711         body:
2712           application/json:
2713             schema: |
2714               {
2715                 "id":
2716                   "http://openinterconnect.org/asamapping/schemas/asa.operation.hvacfanmode.json#",
2717                 "$schema": "http://json-schema.org/draft-04/schema#",
2718                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2719 reserved.",
2720                 "title": "HVAC Fan Mode",
2721                 "definitions": {
2722                   "asa.operation.hvacfanmode": {
2723                     "type": "object",
2724                     "properties": {
2725                       "mode": {
2726                         "type": "integer",
2727                         "description": "Current mode of device.",
2728                         "x-ocf-conversion": {

```

```

2729         "x-ocf-alias": "oic.r.mode",
2730         "x-to-ocf": [
2731             "modearray = [Auto,Circulation,Continuous]",
2732             "ocf.mode[0] = modearray[mode]"
2733         ],
2734         "x-from-ocf": [
2735             "modearray = [Auto,Circulation,Continuous]",
2736             "mode = indexof modeArray[ocf.mode[0]]"
2737         ]
2738     },
2739     },
2740     "supportedmodes": {
2741         "type": "array",
2742         "items": {
2743             "type": "integer"
2744         },
2745         "description": "Array of supported modes",
2746         "x-ocf-conversion": {
2747             "x-ocf-alias": "oic.r.mode",
2748             "x-to-ocf": [
2749                 "modearray = [Auto,Circulation,Continuous]",
2750                 "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
modearray[supportedmodes[x]]"
2751             ],
2752             "x-from-ocf": [
2753                 "modearray = [Auto,Circulation,Continuous]",
2754                 "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
modearray[ocf.supportedmodes[x]]"
2755             ]
2756         }
2757     }
2758 },
2759 },
2760 },
2761 },
2762 },
2763 "type": "object",
2764 "allOf": [
2765     {"$ref": "#/definitions/asa.operation.hvacfanmode"}
2766 ],
2767 "required": [ "mode", "supportedmodes" ]
2768 }
2769
2770 example: |
2771 {
2772     "rt":      ["oic.r.mode"]
2773 }
2774
2775 post:
2776 body:
2777     application/json:
2778
2779     schema: |
2780     {
2781         "id": "http://openinterconnect.org/asamapping/schemas/asa.operation.hvacfanmode.json#",
2782         "$schema": "http://json-schema.org/draft-04/schema#",
2783         "description": "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
2784         "title": "HVAC Fan Mode",
2785         "definitions": {
2786             "asa.operation.hvacfanmode": {
2787                 "type": "object",
2788                 "properties": {
2789                     "mode": {
2790                         "type": "integer",
2791                         "description": "Current mode of device.",
2792                         "x-ocf-conversion": {
2793                             "x-ocf-alias": "oic.r.mode",
2794                             "x-to-ocf": [
2795                                 "modearray = [Auto,Circulation,Continuous]",
2796                                 "ocf.mode[0] = modearray[mode]"

```

```

2797         ],
2798         "x-from-ocf": [
2799             "modearray = [Auto,Circulation,Continuous]",
2800             "mode = indexof modeArray[ocf.mode[0]]"
2801         ]
2802     },
2803 },
2804 "supportedmodes": {
2805     "type": "array",
2806     "items": {
2807         "type": "integer"
2808     },
2809     "description": "Array of supported modes",
2810     "x-ocf-conversion": {
2811         "x-ocf-alias": "oic.r.mode",
2812         "x-to-ocf": [
2813             "modearray = [Auto,Circulation,Continuous]",
2814             "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
2815 modearray[supportedmodes[x]]"
2816         ],
2817         "x-from-ocf": [
2818             "modearray = [Auto,Circulation,Continuous]",
2819             "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
2820 modearray[ocf.supportedmodes[x]]"
2821         ]
2822     }
2823 }
2824 }
2825 }
2826 },
2827 "type": "object",
2828 "allOf": [
2829     {"$ref": "#/definitions/asa.operation.hvacfanmode"}
2830 ],
2831 "required": [ "mode","supportedmodes" ]
2832 }
2833
2834 responses :
2835 200:
2836     body:
2837         application/json:
2838             schema: |
2839                 {
2840                     "id":
2841 "http://openinterconnect.org/asamapping/schemas/asa.operation.hvacfanmode.json#",
2842                     "$schema": "http://json-schema.org/draft-04/schema#",
2843                     "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2844 reserved.",
2845                     "title": "HVAC Fan Mode",
2846                     "definitions": {
2847                         "asa.operation.hvacfanmode": {
2848                             "type": "object",
2849                             "properties": {
2850                                 "mode": {
2851                                     "type": "integer",
2852                                     "description": "Current mode of device.",
2853                                     "x-ocf-conversion": {
2854                                         "x-ocf-alias": "oic.r.mode",
2855                                         "x-to-ocf": [
2856                                             "modearray = [Auto,Circulation,Continuous]",
2857                                             "ocf.mode[0] = modearray[mode]"
2858                                         ],
2859                                         "x-from-ocf": [
2860                                             "modearray = [Auto,Circulation,Continuous]",
2861                                             "mode = indexof modeArray[ocf.mode[0]]"
2862                                         ]
2863                                     }
2864                                 }
2865                             }
2866                         }
2867                     }
2868                 }

```

```

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

```

2895 8.13.5 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.

2896 8.13.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/HvacFanModeResURI		get	post		

2897 8.14 On Off Mapping

2898 8.14.1 Introduction

2899 This API defines the mapping between an instance of an OCF Binary Switch Resource and the
2900 equivalent Interface set by AllJoyn. A discovered instance of a Binary Switch is always mapped
2901 to an Operation.OnOffStatus interface. A RETRIEVE on a Binary Switch maps to an action on an
2902 instance of an Operation.OnOffStatus Interface. An UPDATE on a Binary Switch maps to a method
2903 invocation on either Operation.OnControl or OffControl. value = true maps to Operation.OnControl
2904 value = false maps to Operation.OffControl

```

2905 8.14.2 Example URI
2906 /OnOffResURI
2907 8.14.3 Resource Type
2908 The resource type (rt) is defined as: oic.r.switch.binary.
2909 8.14.4 RAML Definition
2910 #%RAML 0.8
2911 title: OnOffInterfaceMapping
2912 version: OCFv1.0.0-20170317
2913 traits:
2914   - interface-actuator :
2915     queryParameters:
2916       if:
2917         enum: ["oic.if.a", "oic.if.baseline"]
2918   - interface-all :
2919     queryParameters:
2920       if:
2921         enum: ["oic.if.s", "oic.if.a", "oic.if.baseline"]
2922
2923 /OnOffResURI:
2924   description: |
2925     This API defines the mapping between an instance of an OCF Binary Switch Resource and the
2926     equivalent Interface set by AllJoyn
2927     A discovered instance of a Binary Switch is always
2928     mapped to an Operation.OnOffStatus interface.
2929     A RETRIEVE on a Binary Switch maps to an action on an instance of an Operation.OnOffStatus
2930     Interface.
2931     An UPDATE on a Binary Switch maps to a method invocation on either Operation.OnControl or
2932     OffControl.
2933     value = true maps to Operation.OnControl
2934     value = false maps to Operation.OffControl
2935
2936   get:
2937     is : ['interface-all']
2938     responses :
2939       200:
2940         body:
2941           application/json:
2942             schema: |
2943               {
2944                 "id":
2945 "http://openinterconnect.org/asamapping/schemas/asa.operation.onoffstatus.json#",
2946                 "$schema": "http://json-schema.org/draft-04/schema#",
2947                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2948 reserved.",
2949                 "title": "On Off Mapping",
2950                 "definitions": {
2951                   "asa.operation.onoffstatus": {
2952                     "type": "object",
2953                     "properties": {
2954                       "onoff": {
2955                         "type": "boolean",
2956                         "description": "On/Off status of the device",
2957                         "x-ocf-conversion": {
2958                           "x-ocf-alias": "oic.r.switch.binary",
2959                           "x-to-ocf": [
2960                             "value = onoff"
2961                           ],
2962                           "x-from-ocf": [

```



```

2963         "onoff = value"
2964     ]
2965 }
2966 }
2967 }
2968 }
2969 },
2970 "type": "object",
2971 "allOf": [
2972     {"$ref": "#/definitions/asa.operation.onoffstatus"}
2973 ],
2974 "required": [ "onoff" ]
2975 }
2976
2977 example: |
2978     {
2979         "rt":      ["oic.r.switch.binary"]
2980     }
2981
2982 post:
2983     is : ['interface-actuator']
2984     body:
2985         application/json:
2986             schema: |
2987                 {
2988                     "id": "http://openinterconnect.org/asamapping/schemas/asa.operation.oncontrol.json#",
2989                     "$schema": "http://json-schema.org/draft-04/schema#",
2990                     "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2991 reserved.",
2992                     "title": "On/Off Control",
2993                     "definitions": {
2994                         "oic.r.switch.binary": {
2995                             "properties": {
2996                                 "oneOf": [
2997                                     {
2998                                         "properties": {
2999                                             "value": {
3000                                                 "type": "boolean",
3001                                                 "enum": [true]
3002                                             },
3003                                             "x-ocf-conversion": {
3004                                                 "x-x-from-ocf": [
3005                                                     "asa.operation.oncontrol::switchon()"
3006                                                 ]
3007                                             }
3008                                         },
3009                                         },
3010                                     {
3011                                         "properties": {
3012                                             "value": {
3013                                                 "type": "boolean",
3014                                                 "enum": [false]
3015                                             },
3016                                             "x-ocf-conversion": {
3017                                                 "x-x-from-ocf": [
3018                                                     "asa.operation.offcontrol::switchoff()"
3019                                                 ]
3020                                             }
3021                                         },
3022                                     }
3023                                 ]
3024                             },
3025                         },
3026                     "asa.operation.oncontrol": {
3027                         "type": "object",
3028                         "properties": {
3029                             "switchon": {

```

```

3030         "type": "string",
3031         "format": "method",
3032         "description": "Turn on the device",
3033         "x-ocf-conversion": {
3034             "x-ocf-alias": "oic.r.switch.binary",
3035             "x-to-ocf": [
3036                 "value = true"
3037             ]
3038         }
3039     }
3040 }
3041 },
3042 "asa.operation.offcontrol": {
3043     "type": "object",
3044     "properties": {
3045         "switchon": {
3046             "type": "string",
3047             "format": "method",
3048             "description": "Turn off the device",
3049             "x-ocf-conversion": {
3050                 "x-ocf-alias": "oic.r.switch.binary",
3051                 "x-to-ocf": [
3052                     "value = false"
3053                 ]
3054             }
3055         }
3056     }
3057 }
3058 },
3059 "type": "object",
3060 "oneOf": [
3061     {"$ref": "#/definitions/oic.r.switch.binary"},
3062     {"$ref": "#/definitions/asa.operation.oncontrol"},
3063     {"$ref": "#/definitions/asa.operation.offcontrol"}
3064 ]
3065 }
3066
3067 responses :
3068 200:
3069     body:
3070         application/json:
3071             schema: |
3072                 {
3073                     "id":
3074 "http://openinterconnect.org/asamapping/schemas/asa.operation.oncontrol.json#",
3075                     "$schema": "http://json-schema.org/draft-04/schema#",
3076                     "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
3077 reserved.",
3078                     "title": "On/Off Control",
3079                     "definitions": {
3080                         "oic.r.switch.binary": {
3081                             "properties": {
3082                                 "oneOf": [
3083                                     {
3084                                         "properties": {
3085                                             "value": {
3086                                                 "type": "boolean",
3087                                                 "enum": [true]
3088                                             },
3089                                             "x-ocf-conversion": {
3090                                                 "x-x-from-ocf": [
3091                                                     "asa.operation.oncontrol::switchon()"
3092                                                 ]
3093                                             }
3094                                         }
3095                                     },
3096                                     {
3097                                         "properties": {

```

```

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

```

8.14.5 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

8.14.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/OnOffResURI		get	post		

8.15 Oven Cycle Phase Mapping

8.15.1 Introduction

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

8.15.2 Example URI

/OvenCyclePhaseResURI

8.15.3 Resource Type

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

8.15.4 RAML Definition

```
##RAML 0.8
title: OvenCyclePhaseInterfaceMapping
version: OCFv1.0.0-20170317

traits:
  - interface-sensor :
      queryParameters:

        if:
          enum: ["oic.if.s", "oic.if.baseline"]

/OvenCyclePhaseResURI:

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

  is : ['interface-sensor']

  get:
    responses :

      200:
        body:
          application/json:

            schema: |

              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.operation.ovencyclephase.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Oven Cycle Phase",
                "definitions": {
                  "asa.operation.ovencyclephase": {
                    "type": "object",
                    "properties": {
                      "cyclephase": {
                        "type": "integer",
```

```

3211         "description": "Current phase of the operational cycle",
3212         "x-ocf-conversion": {
3213             "x-ocf-alias": "oic.r.operationalstate",
3214             "x-to-ocf": [
3215                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
3216                 "currentmachinestate = phasearray[cyclephase]"
3217             ],
3218             "x-from-ocf": [
3219                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
3220                 "cyclephase = indexof statearray[currentmachinestate[0]]"
3221             ]
3222         }
3223     },
3224     "supportedcyclephases": {
3225         "type": "array",
3226         "items": {
3227             "type": "integer"
3228         },
3229         "description": "Array of cycle phases supported by the Appliance.",
3230         "x-ocf-conversion": {
3231             "x-ocf-alias": "oic.r.operationalstate",
3232             "x-to-ocf": [
3233                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
3234                 "for x=0, x < sizeof(supportedcyclephases): machinestates[x] =
3235 phasearray[supportedcyclephases[x]]"
3236             ],
3237             "x-from-ocf": [
3238                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
3239                 "for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof
3240 phasearray[machinestates[x]]"
3241             ]
3242         }
3243     },
3244     "getvendorphasesdescription": {
3245         "x-ocf-type": "method",
3246         "description": "Get cycle phases description",
3247         "x-ocf-conversion": {
3248             "x-ocf-alias": "oic.r.action"
3249         }
3250     }
3251 }
3252 },
3253 {
3254     "type": "object",
3255     "allOf": [
3256         {"$ref": "#/definitions/asa.operation.ovencyclephase"}
3257     ],
3258     "required": [ "cyclephase","supportedcyclephases" ]
3259 }
3260
3261     example: |
3262     {
3263         "rt":      ["oic.r.operationalstate"]
3264     }
3265

```

3266 8.15.5 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	Array of cycle phas es supp orted

			phasearray[machinestate s[x]]	by the Appli ance.
cyclephase	oic.r.oper ationalsta te	phasearray = [Unavailable,Preheating,Cooki ng,Cleaning]currentmachinestate = phasearray[cyclephase]	phasearray = [Unavailable,Preheating, Cooking,Cleaning]cycleph ase = indexof statearray[currentmachin estate[0]]	Curre nt phas e of the opera tional cycle
getvendorph asedescripti on	oic.r.actio n			Get cycle phas es descr iption

3267 **8.15.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/OvenCyclePhaseResURI		get			

3268

Annex A Swagger2.0 (Informative)

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      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      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      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" }
          }
        }
      }
    }
  },
}
```

```

3329     "post": {
3330         "description": "",
3331         "parameters": [
3332             { "$ref": "#/parameters/interface-all"},
3333             {
3334                 "name": "body",
3335                 "in": "body",
3336                 "required": true,
3337                 "schema": { "$ref": "#/definitions/UpdateSchema" }
3338             }
3339         ],
3340         "responses": {
3341             "200": {
3342                 "description": "",
3343                 "schema": { "$ref": "#/definitions/UpdateSchema" }
3344             }
3345         }
3346     }
3347 },
3348 },
3349 "parameters": {
3350     "interface-all" : {
3351         "in" : "query",
3352         "name" : "if",
3353         "type" : "string",
3354         "enum" : ["oic.if.a", "oic.if.baseline"]
3355     }
3356 },
3357 "definitions": {
3358     "RetrieveSchema" :
3359     {
3360         "properties": {
3361             "maxvolume": {
3362                 "type": "integer",
3363                 "x-ocf-conversion": {
3364                     "x-from-ocf": [
3365                         "maxvolume = range[1]",
3366                         "otherwise: maxvalue = 100"
3367                     ],
3368                     "x-ocf-alias": "oic.r.audio",
3369                     "x-to-ocf": [
3370                         "range[0] = 0",
3371                         "range[1] = maxvolume"
3372                     ]
3373                 }
3374             },
3375             "mute": {
3376                 "type": "boolean",
3377                 "x-ocf-conversion": {
3378                     "x-from-ocf": [
3379                         "mute = ocf.mute"
3380                     ],
3381                     "x-ocf-alias": "oic.r.audio",
3382                     "x-to-ocf": [
3383                         "ocf.mute = mute"
3384                     ]
3385                 }
3386             },
3387             "volume": {
3388                 "description": "Speaker volume index",
3389                 "type": "integer",
3390                 "x-ocf-conversion": {
3391                     "x-from-ocf": [
3392                         "volume = ocf.volume"
3393                     ],
3394                     "x-ocf-alias": "oic.r.audio",
3395                     "x-to-ocf": [
3396                         "ocf.volume = volume"
3397                     ]
3398                 }
3399             }
3400         }
3401     }
3402 }

```



```

3400     },
3401     "required": [
3402         "volume",
3403         "maxvolume",
3404         "mute"
3405     ],
3406     "type": "object"
3407 }
3408
3409 ,
3410 "UpdateSchema" :
3411 {
3412     "properties": {
3413         "maxvolume": {
3414             "type": "integer",
3415             "x-ocf-conversion": {
3416                 "x-from-ocf": [
3417                     "maxvolume = range[1]",
3418                     "otherwise: maxvalue = 100"
3419                 ],
3420                 "x-ocf-alias": "oic.r.audio",
3421                 "x-to-ocf": [
3422                     "range[0] = 0",
3423                     "range[1] = maxvolume"
3424                 ]
3425             }
3426         },
3427         "mute": {
3428             "type": "boolean",
3429             "x-ocf-conversion": {
3430                 "x-from-ocf": [
3431                     "mute = ocf.mute"
3432                 ],
3433                 "x-ocf-alias": "oic.r.audio",
3434                 "x-to-ocf": [
3435                     "ocf.mute = mute"
3436                 ]
3437             }
3438         },
3439         "volume": {
3440             "description": "Speaker volume index",
3441             "type": "integer",
3442             "x-ocf-conversion": {
3443                 "x-from-ocf": [
3444                     "volume = ocf.volume"
3445                 ],
3446                 "x-ocf-alias": "oic.r.audio",
3447                 "x-to-ocf": [
3448                     "ocf.volume = volume"
3449                 ]
3450             }
3451         }
3452     },
3453     "required": [
3454         "volume",
3455         "maxvolume",
3456         "mute"
3457     ],
3458     "type": "object"
3459 }
3460
3461 }
3462
3463

```

3464 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] = maxvolume range[1] = maxvolume	range[1] otherwise: maxvalue = 100	
mute	oic.r.audio	ocf.mute = mute	mute = ocf.mute	

A.1.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/AudioVolumeResURI		get	post		

A.2 Climate Control Mode Mapping

A.2.1 Introduction

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: mode, supportedmodes -> Mode Resource operationalstate -> OperationalState Resource. This can be represented in OCF either as two distinct Resource instances or a single instance with two RTs (oic.r.mode, oic.r.operationalstate)

A.2.2 Example URI

/ClimateControlModeResURI

A.2.3 Resource Type

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

A.2.4 Swagger2.0 Definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Climate Control Mode 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. \\\nAS IS\\n 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      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      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": {
    "/ClimateControlModeResURI" : {
      "get": {
        "description": "This API defines the mapping between an instance of an AllJoyn
ClimateControlMode interface and the OCF equivalent Resource.\nClimateControlMode has three
Properties; these map as follows:\n mode, supportedmodes -> Mode Resource\n operationalstate ->
```

```

3514 OperationalState Resource\nThis can be represented in OCF either as two distinct Resource instances
3515 or a single instance with two RTs (oic.r.mode, oic.r.operationalstate)\n",
3516     "parameters": [
3517         {"$ref": "#/parameters/interface-actuator"}
3518     ],
3519     "responses": {
3520         "200": {
3521             "description": "",
3522             "x-example":
3523                 {
3524                     "rt": ["oic.r.mode", "oic.operationalstate"]
3525                 },
3526             "schema": { "$ref": "#/definitions/RetrieveSchema" }
3527         }
3528     },
3529 },
3530 },
3531 "post": {
3532     "description": "",
3533     "parameters": [
3534         {"$ref": "#/parameters/interface-actuator"},
3535         {
3536             "name": "body",
3537             "in": "body",
3538             "required": true,
3539             "schema": { "$ref": "#/definitions/UpdateSchema" }
3540         }
3541     ],
3542     "responses": {
3543         "200": {
3544             "description": "",
3545             "schema": { "$ref": "#/definitions/UpdateSchema" }
3546         }
3547     }
3548 },
3549 },
3550 },
3551 "parameters": {
3552     "interface-actuator" : {
3553         "in" : "query",
3554         "name" : "if",
3555         "type" : "string",
3556         "enum" : ["oic.if.a", "oic.if.baseline"]
3557     }
3558 },
3559 "definitions": {
3560     "RetrieveSchema" :
3561     {
3562         "properties": {
3563             "mode": {
3564                 "description": "Current mode of device.",
3565                 "type": "integer",
3566                 "x-ocf-conversion": {
3567                     "x-from-ocf": [
3568                         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3569                         "mode = indexof modeArray[ocf.mode[0]]"
3570                     ],
3571                     "x-ocf-alias": "oic.r.mode",
3572                     "x-to-ocf": [
3573                         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3574                         "ocf.mode[0] = modearray[mode]"
3575                     ]
3576                 }
3577             },
3578             "operationalstate": {
3579                 "description": "Current status of device",
3580                 "type": "integer",
3581                 "x-ocf-conversion": {
3582                     "x-from-ocf": [
3583                         "statearray = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
3584                         "operationalstate = indexof statearray[currentmachinestate[0]]"

```

```

3585         ],
3586         "x-ocf-alias": "oic.r.operationalstate",
3587         "x-to-ocf": [
3588             "machinestates = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
3589             "currentmachinestate = machinestates[operationalstate]"
3590         ]
3591     },
3592 },
3593 "supportedmodes": {
3594     "description": "Array of supported modes",
3595     "items": {
3596         "type": "integer"
3597     },
3598     "type": "array",
3599     "x-ocf-conversion": {
3600         "x-from-ocf": [
3601             "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3602             "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
3603 modearray[ocf.supportedmodes[x]]"
3604         ],
3605         "x-ocf-alias": "oic.r.mode",
3606         "x-to-ocf": [
3607             "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3608             "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
3609 modearray[supportedmodes[x]]"
3610         ]
3611     }
3612 },
3613 },
3614 "required": [
3615     "mode",
3616     "supportedmodes",
3617     "operationalstate"
3618 ],
3619 "type": "object"
3620 }
3621
3622 /
3623 "UpdateSchema" :
3624 {
3625     "properties": {
3626         "mode": {
3627             "description": "Current mode of device.",
3628             "type": "integer",
3629             "x-ocf-conversion": {
3630                 "x-from-ocf": [
3631                     "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3632                     "mode = indexof modeArray[ocf.mode[0]]"
3633                 ],
3634                 "x-ocf-alias": "oic.r.mode",
3635                 "x-to-ocf": [
3636                     "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3637                     "ocf.mode[0] = modearray[mode]"
3638                 ]
3639             }
3640         },
3641         "operationalstate": {
3642             "description": "Current status of device",
3643             "type": "integer",
3644             "x-ocf-conversion": {
3645                 "x-from-ocf": [
3646                     "statearray = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
3647                     "operationalstate = indexof statearray[currentmachinestate[0]]"
3648                 ],
3649                 "x-ocf-alias": "oic.r.operationalstate",
3650                 "x-to-ocf": [
3651                     "machinestates = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
3652                     "currentmachinestate = machinestates[operationalstate]"
3653                 ]
3654             }
3655         }
3656     }
3657 }

```

```

3656         "supportedmodes": {
3657             "description": "Array of supported modes",
3658             "items": {
3659                 "type": "integer"
3660             },
3661             "type": "array",
3662             "x-ocf-conversion": {
3663                 "x-from-ocf": [
3664                     "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3665                     "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
3666 modearray[ocf.supportedmodes[x]]"
3667                 ],
3668                 "x-ocf-alias": "oic.r.mode",
3669                 "x-to-ocf": [
3670                     "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3671                     "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
3672 modearray[supportedmodes[x]]"
3673                 ]
3674             }
3675         },
3676     },
3677     "required": [
3678         "mode",
3679         "supportedmodes",
3680         "operationalstate"
3681     ],
3682     "type": "object"
3683 }
3684
3685 }
3686 }
3687

```

A.2.5 Property Definition

['AIJ 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.

A.2.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/ClimateControlModeResURI		get	post		

A.3 Closed Status Mapping

A.3.1 Introduction

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

A.3.2 Example URI

/ClosedStatusResURI

A.3.3 Resource Type

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

A.3.4 Swagger2.0 Definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Closed Status 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      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      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      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": {
    "/ClosedStatusResURI" : {
      "get": {
        "description": "This API defines the mapping between an instance of an AllJoyn ClosedStatus
Interface and\nthe OCF Door Resource.\n",
        "parameters": [
          { "$ref": "#/parameters/interface-all" }
        ],
        "responses": {
          "200": {
            "description": "",
            "x-example": {
              "rt": ["oic.r.door"]
            },
            "schema": { "$ref": "#/definitions/RetrieveSchema" }
          }
        }
      }
    }
  },
  "parameters": {
    "interface-all" : {
      "in" : "query",
      "name" : "if",

```

```

3753         "type" : "string",
3754         "enum" : ["oic.if.s", "oic.if.baseline"]
3755     },
3756 },
3757 "definitions": {
3758     "RetrieveSchema" :
3759     {
3760         "properties": {
3761             "isclosed": {
3762                 "description": "Open/Closed status Indicator",
3763                 "type": "boolean",
3764                 "x-ocf-conversion": {
3765                     "x-from-ocf": [
3766                         "isClosed = (openState == Closed)"
3767                     ],
3768                     "x-ocf-alias": "oic.r.door",
3769                     "x-to-ocf": [
3770                         "if isClosed ocf.openState = Closed.",
3771                         "if !isClosed ocf.openState = Open."
3772                     ]
3773                 }
3774             },
3775             "required": [
3776                 "isclosed"
3777             ],
3778             "type": "object"
3779         }
3780     }
3781 }
3782 }
3783 }
3784

```

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

A.3.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/ClosedStatusResURI		get			

A.4 Air Quality Mapping

A.4.1 Introduction

This API defines the mapping between the AllJoyn AirQuality interface and the OCF AirQuality Resource.

If more than one instance of the AirQuality interface is exposed then each instance maps to an instance of the OCF AirQuality Resource.

The mapping defined in the schema describes the population of the OCF AirQuality Resource. Even if there is only a single instance of an OCF AirQuality Resource this shall be included in an instance of an OCF AirQualityCollection.

The number of links in the collection equates to the number of instances of the AllJoyn CurrentAirQuality interface that are exposed.

When mapping from OCF the valueType of the Resource shall be introspected, this API is invoked only if this is set to 'Measured'

A.4.2 Example URI

/CurrentAirQualityResURI

A.4.3 Resource Type

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

A.4.4 Swagger2.0 Definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Air Quality 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      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      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": {
    "/CurrentAirQualityResURI" : {
      "get": {
        "description": "This API defines the mapping between the AllJoyn AirQuality interface and
the OCF AirQuality Resource.\nIf more than one instance of the AirQuality interface is exposed then
each instance maps to an instance of the OCF AirQuality Resource.\nThe mapping defined in the
schema describes the population of the OCF AirQuality Resource.\nEven if there is only a single
instance of an OCF AirQuality Resource this shall be included in an instance of an OCF
AirQualityCollection.\nThe number of links in the collection equates to the number of instances of
the AllJoyn CurrentAirQuality interface that are exposed.\nWhen mapping from OCF the valueType of
the Resource shall be introspected, this API is invoked only if this is set to 'Measured'\n",
        "parameters": [
          { "$ref": "#/parameters/interface-sensor" }
        ],
        "responses": {
          "200": {
            "description": "",
            "x-example": {
              "rt": ["oic.r.airqualitycollection"]
            },
            "schema": { "$ref": "#/definitions/RetrieveSchema" }
          }
        }
      }
    }
  },
  "parameters": {
    "interface-sensor" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.s", "oic.if.baseline"]
    }
  }
}
```



```

3868 },
3869 "definitions": {
3870   "RetrieveSchema" :
3871     {
3872       "properties": {
3873         "contaminanttype": {
3874           "description": "The contaminant type",
3875           "type": "integer",
3876           "x-ocf-conversion": {
3877             "x-from-ocf": [
3878               "contaminanttype = indexof contaminanttypearray[ocf.contaminanttype]"
3879             ],
3880             "x-ocf-alias": "oic.r.airquality",
3881             "x-to-ocf": [
3882               "valuetype = Measured",
3883               "contaminanttypearray = [CH2O,CO2,CO,PM2_5,PM10,VOC]",
3884               "ocf.contaminanttype = contaminanttypearray[contaminanttype]"
3885             ]
3886           }
3887         },
3888         "currentvalue": {
3889           "type": "number",
3890           "x-ocf-conversion": {
3891             "x-from-ocf": [
3892               "currentvalue = contaminantvalue"
3893             ],
3894             "x-ocf-alias": "oic.r.airquality",
3895             "x-to-ocf": [
3896               "contaminantvalue = currentvalue"
3897             ]
3898           }
3899         },
3900         "maxvalue": {
3901           "type": "number",
3902           "x-ocf-conversion": {
3903             "x-from-ocf": [
3904               "maxvalue = range[1]"
3905             ],
3906             "x-ocf-alias": "oic.r.airquality",
3907             "x-to-ocf": [
3908               "range[1] = maxvalue"
3909             ]
3910           }
3911         },
3912         "minvalue": {
3913           "type": "number",
3914           "x-ocf-conversion": {
3915             "x-from-ocf": [
3916               "minvalue = range[0]"
3917             ],
3918             "x-ocf-alias": "oic.r.airquality",
3919             "x-to-ocf": [
3920               "range[0] = minvalue"
3921             ]
3922           }
3923         },
3924         "precision": {
3925           "type": "number",
3926           "x-ocf-conversion": {
3927             "x-from-ocf": [
3928               "precision = ocf.precision"
3929             ],
3930             "x-ocf-alias": "oic.r.airquality",
3931             "x-to-ocf": [
3932               "ocf.precision = precision"
3933             ]
3934           }
3935         },
3936         "updatemintime": {
3937           "type": "integer",
3938           "x-ocf-conversion": {

```

```

3939         "x-from-ocf": [
3940             "updatemintime = ocf.minnotifyperiod"
3941         ],
3942         "x-ocf-alias": "oic.r.value.conditional",
3943         "x-to-ocf": [
3944             "ocf.minnotifyperiod = updatemintime"
3945         ]
3946     }
3947 },
3948 },
3949     "required": [
3950         "contaminanttype",
3951         "currentvalue",
3952         "minvalue",
3953         "maxvalue",
3954         "precision",
3955         "updatemintime"
3956     ],
3957     "type": "object"
3958 }
3959 }
3960 }
3961 }
3962

```

3963 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,VOC]ocf.contaminanttype = contaminanttypearray[contaminanttype]	contaminanttype = indexof contaminanttypearray[ocf.contaminanttype]	The contaminant type

3964 A.4.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentAirQualityResURI		get			

3965 A.5 Air Quality Level Mapping

3966 A.5.1 Introduction

3967 This API defines the mapping between the AllJoyn AirQualityLevel interface and the OCF
3968 AirQuality Resource.
3969 If more than one instance of the AirQualityLevel interface is exposed then each instance maps to
3970 an instance of the OCF AirQuality Resource.
3971 The mapping defined in the schema describes the population of the OCF AirQuality Resource.

3972 Even if there is only a single instance of an OCF AirQuality Resource then this shall be included
3973 in an instance of an OCF AirQualityCollection.
3974 The number of links in the collection equates to the number of instances of the AllJoyn
3975 CurrentAirQuality interface that are exposed.
3976 When mapping from OCF the valueType of the Resource shall be introspected, this API is invoked
3977 only if this is set to 'Qualitative'
3978

3979 **A.5.2 Example URI**

3980 /CurrentAirQualityLevelResURI

3981 **A.5.3 Resource Type**

3982 The resource type (rt) is defined as: ['oic.r.airqualitycollection'].

3983 **A.5.4 Swagger2.0 Definition**

```
3984 {  
3985   "swagger": "2.0",  
3986   "info": {  
3987     "title": "Air Quality Level Mapping",  
3988     "version": "OCFv1.0.0-20170317",  
3989     "license": {  
3990       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",  
3991       "x-description": "Redistribution and use in source and binary forms, with or without  
3992 modification, are permitted provided that the following conditions are met:\n      1.  
3993 Redistributions of source code must retain the above copyright notice, this list of conditions and  
3994 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above  
3995 copyright notice, this list of conditions and the following disclaimer in the documentation and/or  
3996 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open  
3997 Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT  
3998 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR  
3999 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n\n      IN NO EVENT SHALL THE Open Connectivity  
4000 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,  
4001 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
4002 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n\n      HOWEVER CAUSED AND  
4003 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR  
4004 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY  
4005 OF SUCH DAMAGE.\n\n    }  
4006   },  
4007 },  
4008 "schemes": ["http"],  
4009 "consumes": ["application/json"],  
4010 "produces": ["application/json"],  
4011 "paths": {  
4012   "/CurrentAirQualityLevelResURI" : {  
4013     "get": {  
4014       "description": "This API defines the mapping between the AllJoyn AirQualityLevel interface  
4015 and the OCF AirQuality Resource.\nIf more than one instance of the AirQualityLevel interface is  
4016 exposed then each instance maps to an instance of the OCF AirQuality Resource.\n\nThe mapping defined  
4017 in the schema describes the population of the OCF AirQuality Resource.\n\nEven if there is only a  
4018 single instance of an OCF AirQuality Resource then this shall be included in an instance of an OCF  
4019 AirQualityCollection.\n\nThe number of links in the collection equates to the number of instances of  
4020 the AllJoyn CurrentAirQuality interface that are exposed.\n\nWhen mapping from OCF the valueType of  
4021 the Resource shall be introspected, this API is invoked only if this is set to 'Qualitative'\n",  
4022       "parameters": [  
4023         {"$ref": "#/parameters/interface-sensor"}  
4024       ],  
4025       "responses": {  
4026         "200": {  
4027           "description": "",  
4028           "x-example": {  
4029             "rt": ["oic.r.airqualitycollection"]  
4030           }  
4031         },  
4032         "schema": { "$ref": "#/definitions/RetrieveSchema" }  
4033       }  
4034     }  
4035   }
```

```

4036     }
4037   }
4038 },
4039 "parameters": {
4040   "interface-sensor" : {
4041     "in" : "query",
4042     "name" : "if",
4043     "type" : "string",
4044     "enum" : ["oic.if.s", "oic.if.baseline"]
4045   }
4046 },
4047 "definitions": {
4048   "RetrieveSchema" :
4049   {
4050     "properties": {
4051       "contaminanttype": {
4052         "description": "The contaminant type",
4053         "type": "integer",
4054         "x-ocf-conversion": {
4055           "x-from-ocf": [
4056             "if ocf.contaminanttype = CH2O, contaminanttype = 0",
4057             "if ocf.contaminanttype = CO2, contaminanttype = 1",
4058             "if ocf.contaminanttype = CO, contaminanttype = 2",
4059             "if ocf.contaminanttype = PM2_5, contaminanttype = 3",
4060             "if ocf.contaminanttype = PM10, contaminanttype = 4",
4061             "if ocf.contaminanttype = VOC, contaminanttype = 5",
4062             "if ocf.contaminanttype = Smoke, contaminanttype = 253",
4063             "if ocf.contaminanttype = Odor, contaminanttype = 254",
4064             "if ocf.contaminanttype = AirPollution, contaminanttype = 255"
4065           ],
4066           "x-ocf-alias": "oic.r.airquality",
4067           "x-to-ocf": [
4068             "valuetype = Qualitative",
4069             "if contaminanttype = 0, ocf.contaminanttype = CH2O",
4070             "if contaminanttype = 1, ocf.contaminanttype = CO2",
4071             "if contaminanttype = 2, ocf.contaminanttype = CO",
4072             "if contaminanttype = 3, ocf.contaminanttype = PM2_5",
4073             "if contaminanttype = 4, ocf.contaminanttype = PM10",
4074             "if contaminanttype = 5, ocf.contaminanttype = VOC",
4075             "if contaminanttype = 253, ocf.contaminanttype = Smoke",
4076             "if contaminanttype = 254, ocf.contaminanttype = Odor",
4077             "if contaminanttype = 255, ocf.contaminanttype = AirPollution"
4078           ]
4079         }
4080       },
4081       "currentlevel": {
4082         "type": "integer",
4083         "x-ocf-conversion": {
4084           "x-from-ocf": [
4085             "currentlevel = contaminantvalue"
4086           ],
4087           "x-ocf-alias": "oic.r.airquality",
4088           "x-to-ocf": [
4089             "contaminantvalue = currentlevel"
4090           ]
4091         }
4092       },
4093       "maxlevel": {
4094         "type": "integer",
4095         "x-ocf-conversion": {
4096           "x-from-ocf": [
4097             "maxvalue = range[1]"
4098           ],
4099           "x-ocf-alias": "oic.r.airquality",
4100           "x-to-ocf": [
4101             "range[0] = 0",
4102             "range[1] = maxvalue"
4103           ]
4104         }
4105       }
4106     },

```

```

4107     "required": [
4108         "contaminanttype",
4109         "currentlevel",
4110         "maxlevel"
4111     ],
4112     "type": "object"
4113 }
4114
4115 }
4116 }
4117

```

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

4119 A.5.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentAirQualityLevelResURI		get			

4120 A.6 Current Humidity Mapping

4121 A.6.1 Introduction

4122 This API defines the mapping between an instance of an OCF Humidity which exposes only a
 4123 sensor interface
 4124 and the AllJoyn Current Humidity interface.
 4125 A RETRIEVE on a Temperature Sensor maps to an action on an instance of an
 4126 Environment.CurrentTemperature Interface.
 4127

4128 A.6.2 Example URI

4129 /CurrentHumidityResURI

4130 A.6.3 Resource Type

4131 The resource type (rt) is defined as: ['oic.r.humidity'].

4132 A.6.4 Swagger2.0 Definition

```

4133 {
4134   "swagger": "2.0",
4135   "info": {
4136     "title": "Current Humidity Mapping",
4137     "version": "OCFv1.0.0-20170317",
4138     "license": {
4139       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
4140       "x-description": "Redistribution and use in source and binary forms, with or without
4141 modification, are permitted provided that the following conditions are met:\n      1.
4142 Redistributions of source code must retain the above copyright notice, this list of conditions and
4143 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
4144 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
4145 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
4146 Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
4147 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
4148 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
4149 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
4150 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
4151 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND
4152 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
4153 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
4154 OF SUCH DAMAGE.\n"
4155     },
4156   },
4157   "schemes": ["http"],
4158   "consumes": ["application/json"],
4159   "produces": ["application/json"],
4160   "paths": {
4161     "/CurrentHumidityResURI" : {
4162       "get": {
4163         "description": "This API defines the mapping between an instance of an OCF Humidity which
4164 exposes only a sensor interface\nand the AllJoyn Current Humidity interface.\nA RETRIEVE on a
4165 Temperature Sensor maps to an action on an instance of an Environment.CurrentTemperature
4166 Interface.\n",
4167         "parameters": [
4168           {"$ref": "#/parameters/interface-sensor"}
4169         ],
4170         "responses": {
4171           "200": {
4172             "description": "",
4173             "x-example": {
4174               "rt": ["oic.r.humidity"]
4175             }
4176           }
4177         }
4178       }
4179     }
4180   }
4181 }
```

```

4177         ,
4178         "schema": { "$ref": "#/definitions/RetrieveSchema" }
4179     }
4180 }
4181 }
4182 }
4183 },
4184 "parameters": {
4185     "interface-sensor" : {
4186         "in" : "query",
4187         "name" : "if",
4188         "type" : "string",
4189         "enum" : ["oic.if.s", "oic.if.baseline"]
4190     }
4191 },
4192 "definitions": {
4193     "RetrieveSchema" :
4194     {
4195         "properties": {
4196             "currentvalue": {
4197                 "description": "Measured value",
4198                 "type": "number",
4199                 "x-ocf-conversion": {
4200                     "x-from-ocf": [
4201                         "currentvalue = humidity"
4202                     ],
4203                     "x-ocf-alias": "oic.r.humidity",
4204                     "x-to-ocf": [
4205                         "humidity = currentValue"
4206                     ]
4207                 }
4208             },
4209             "maxvalue": {
4210                 "description": "Max measured value for humidty",
4211                 "type": "number",
4212                 "x-ocf-conversion": {
4213                     "x-from-ocf": [
4214                         "maxvalue = range[1]"
4215                     ],
4216                     "x-ocf-alias": "oic.r.humidity",
4217                     "x-to-ocf": [
4218                         "range[0] = 0",
4219                         "range[1] = maxvalue"
4220                     ]
4221                 }
4222             }
4223         },
4224         "required": [
4225             "currentvalue",
4226             "maxvalue"
4227         ],
4228         "type": "object"
4229     }
4230 }
4231 }
4232 }
4233

```

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] 0range[1] maxvalue	maxvalue range[1]	Max measured value for humidty

4235 A.6.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentHumidityResURI		get			

4236 A.7 Current Temperature Mapping

4237 A.7.1 Introduction

4238 This API defines the mapping between an instance of an OCF Temperature which exposes only a
 4239 sensor interface
 4240 and the AllJoyn Current Temperature interface.
 4241 A RETRIEVE on a Temperature Sensor maps to an action on an instance of an
 4242 Environment.CurrentTemperature Interface.
 4243

4244 A.7.2 Example URI

4245 /CurrentTemperatureResURI

4246 A.7.3 Resource Type

4247 The resource type (rt) is defined as: ['oic.r.temperature'].

4248 A.7.4 Swagger2.0 Definition

```

4249 {
4250   "swagger": "2.0",
4251   "info": {
4252     "title": "Current Temperature Mapping",
4253     "version": "OCFv1.0.0-20170317",
4254     "license": {
4255       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
4256       "x-description": "Redistribution and use in source and binary forms, with or without
4257 modification, are permitted provided that the following conditions are met:\n      1.
4258 Redistributions of source code must retain the above copyright notice, this list of conditions and
4259 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
4260 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
4261 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
4262 Connectivity Foundation, INC. \
4263 \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
4264 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
4265 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
4266 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
4267 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
4268 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND
4269 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
4270 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
4271 OF SUCH DAMAGE.\n"
4272     },
4273   "schemes": ["http"],
4274   "consumes": ["application/json"],
4275   "produces": ["application/json"],
4276   "paths": {
4277     "/CurrentTemperatureResURI" : {
4278       "get": {
4279         "description": "This API defines the mapping between an instance of an OCF Temperature
4280 which exposes only a sensor interface\nand the AllJoyn Current Temperature interface.\nA RETRIEVE
4281 on a Temperature Sensor maps to an action on an instance of an Environment.CurrentTemperature
4282 Interface.\n",
4283         "parameters": [
4284           {"$ref": "#/parameters/interface-sensor"}
4285         ],
4286         "responses": {
4287           "200": {
4288             "description": "",
4289             "x-example": {
4290               "rt": ["oic.r.temperature"]
4291             }
4292           }
4293         }
4294       }
4295     }
4296   }
4297 }
```



```

4293         ,
4294         "schema": { "$ref": "#/definitions/RetrieveSchema" }
4295     }
4296 }
4297 }
4298 }
4299 },
4300 "parameters": {
4301     "interface-sensor" : {
4302         "in" : "query",
4303         "name" : "if",
4304         "type" : "string",
4305         "enum" : ["oic.if.s", "oic.if.baseline"]
4306     }
4307 },
4308 "definitions": {
4309     "RetrieveSchema" :
4310     {
4311         "properties": {
4312             "currentvalue": {
4313                 "description": "Measured value",
4314                 "type": "number",
4315                 "x-ocf-conversion": {
4316                     "x-from-ocf": {
4317                         "oneOf": [
4318                             {
4319                                 "properties": {
4320                                     "enum": [
4321                                         "C"
4322                                     ],
4323                                     "units": "string"
4324                                 },
4325                                 "x-from-ocf": [
4326                                     "currentvalue = temperature"
4327                                 ]
4328                             },
4329                             {
4330                                 "properties": {
4331                                     "enum": [
4332                                         "F"
4333                                     ],
4334                                     "units": "string"
4335                                 },
4336                                 "x-from-ocf": [
4337                                     "currentvalue = (temperature-32)*5/9"
4338                                 ]
4339                             },
4340                             {
4341                                 "properties": {
4342                                     "enum": [
4343                                         "K"
4344                                     ],
4345                                     "units": "string"
4346                                 },
4347                                 "x-from-ocf": [
4348                                     "currentvalue = temperature-273.15"
4349                                 ]
4350                             }
4351                         ]
4352                     },
4353                     "x-ocf-alias": "oic.r.temperature",
4354                     "x-to-ocf": [
4355                         "temperature = currentValue",
4356                         "units = C"
4357                     ]
4358                 }
4359             },
4360             "precision": {
4361                 "type": "number",
4362                 "x-ocf-conversion": {
4363                     "x-from-ocf": [

```

```

4364         "precision = ocf.precision"
4365     ],
4366     "x-ocf-alias": "oic.r.temperature",
4367     "x-to-ocf": [
4368         "ocf.precision = precision"
4369     ]
4370 },
4371 },
4372 "updatemintime": {
4373     "type": "integer",
4374     "x-ocf-conversion": {
4375         "x-from-ocf": [
4376             "updatemintime = ocf.minnotifyperiod"
4377         ],
4378         "x-ocf-alias": "oic.r.value.conditional",
4379         "x-to-ocf": [
4380             "ocf.minnotifyperiod = updatemintime"
4381         ]
4382     }
4383 },
4384 },
4385 "required": [
4386     "currentvalue",
4387     "precision",
4388     "updatemintime"
4389 ],
4390 "type": "object"
4391 }
4392 }
4393 }
4394 }
4395

```

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.conditional	ocf.minnotifyperiod = updatemintime	updatemintime = ocf.minnotifyperiod	
precision	oic.r.temperature	ocf.precision = precision	precision = ocf.precision	

A.7.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentTemperatureResURI		get			

A.8 Cycle Control Mapping

A.8.1 Introduction

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

A.8.2 Example URI

/CycleControlResURI

4408 A.8.3 Resource Type

4409 The resource type (rt) is defined as: ['oic.r.operationalstate'].

4410 A.8.4 Swagger2.0 Definition

```
4411 {
4412   "swagger": "2.0",
4413   "info": {
4414     "title": "Cycle Control Mapping",
4415     "version": "OCFv1.0.0-20170317",
4416     "license": {
4417       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
4418       "x-description": "Redistribution and use in source and binary forms, with or without
4419 modification, are permitted provided that the following conditions are met:\n      1.
4420 Redistributions of source code must retain the above copyright notice, this list of conditions and
4421 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
4422 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
4423 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
4424 Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
4425 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
4426 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
4427 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
4428 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
4429 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND
4430 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
4431 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
4432 OF SUCH DAMAGE.\n"
4433     }
4434   },
4435   "schemes": ["http"],
4436   "consumes": ["application/json"],
4437   "produces": ["application/json"],
4438   "paths": {
4439     "/CycleControlResURI" : {
4440       "get": {
4441         "description": "This API defines the mapping between an instance of an AllJoyn CycleControl
4442 interface and the OCF OperationalState Resource.\nThe AllJoyn interface also supports a Method,
4443 ExecuteOperationalCommand; this is handled in OCF using an instance of oic.r.actuator within an
4444 oic.r.action collection.\nPlease see Section 8 of the Mapping Specification for specifics.\n",
4445         "parameters": [
4446           { "$ref": "#/parameters/interface-sensor" }
4447         ],
4448         "responses": {
4449           "200": {
4450             "description": "",
4451             "x-example": {
4452               "rt": ["oic.r.operationalstate"]
4453             },
4454             "schema": { "$ref": "#/definitions/RetrieveSchema" }
4455           }
4456         }
4457       }
4458     }
4459   },
4460 },
4461 {
4462   "parameters": {
4463     "interface-sensor" : {
4464       "in" : "query",
4465       "name" : "if",
4466       "type" : "string",
4467       "enum" : ["oic.if.s", "oic.if.baseline"]
4468     }
4469   },
4470   "definitions": {
4471     "RetrieveSchema" : {
4472       "properties": {
4473         "cyclephase": {
4474           "description": "Current phase of the operational cycle",
4475           "type": "integer",
4476
```

```

4477         "x-ocf-conversion": {
4478             "x-from-ocf": [
4479                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
4480                 "cyclephase = indexof statearray[currentmachinestate[0]]"
4481             ],
4482             "x-ocf-alias": "oic.r.operationalstate",
4483             "x-to-ocf": [
4484                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
4485                 "currentmachinestate = phasearray[cyclephase]"
4486             ]
4487         },
4488     },
4489     "getvendorphasesdescription": {
4490         "description": "Get cycle phases description",
4491         "x-ocf-conversion": {
4492             "x-ocf-alias": "oic.r.action"
4493         },
4494         "x-ocf-type": "method"
4495     },
4496     "supportedcyclephases": {
4497         "description": "Array of cycle phases supported by the Appliance.",
4498         "items": {
4499             "type": "integer"
4500         },
4501         "type": "array",
4502         "x-ocf-conversion": {
4503             "x-from-ocf": [
4504                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
4505                 "for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof
4506 phasearray[machinestates[x]]"
4507             ],
4508             "x-ocf-alias": "oic.r.operationalstate",
4509             "x-to-ocf": [
4510                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
4511                 "for x=0, x < sizeof(supportedcyclephases): machinestates[x] =
4512 phasearray[supportedcyclephases[x]]"
4513             ]
4514         }
4515     },
4516 },
4517 "required": [
4518     "cyclephase",
4519     "supportedcyclephases"
4520 ],
4521 "type": "object"
4522 }
4523 }
4524 }
4525 }
4526

```

A.8.5 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

A.8.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CycleControlResURI		get			

A.9 Fan Speed Level Mapping

A.9.1 Introduction

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

A.9.2 Example URI

/FanSpeedLevelResURI

A.9.3 Resource Type

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

A.9.4 Swagger2.0 Definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Fan Speed Level 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"
    }
  },
}
```

```

4567     "schemes": ["http"],
4568     "consumes": ["application/json"],
4569     "produces": ["application/json"],
4570     "paths": {
4571         "/FanSpeedLevelResURI" : {
4572             "get": {
4573                 "description": "This API defines the mapping between an instance of an AllJoyn
4574 FanSpeedLevel interface and an OCF AirFlow Resource.\nNote that the setting of the FanSpeedLevel to
4575 '0x00' (off) is handled via the 'OffControl' interface rather than writing directly to this
4576 interface.\nIn such a case an instance of Binary Switch shall be exposed on the OCF side; this can
4577 be modeled as AirFlowControl which is then a collection of Binary Switch and AirFlow.\n",
4578                 "parameters": [
4579                     {"$ref": "#/parameters/interface-actuator"}
4580                 ],
4581                 "responses": {
4582                     "200": {
4583                         "description": "",
4584                         "x-example": {
4585                             {
4586                                 "rt": ["oic.r.airflow"]
4587                             }
4588                         },
4589                         "schema": { "$ref": "#/definitions/RetrieveSchema" }
4590                     }
4591                 }
4592             },
4593             "post": {
4594                 "description": "",
4595                 "parameters": [
4596                     {"$ref": "#/parameters/interface-actuator"},
4597                     {
4598                         "name": "body",
4599                         "in": "body",
4600                         "required": true,
4601                         "schema": { "$ref": "#/definitions/UpdateSchema" }
4602                     }
4603                 ],
4604                 "responses": {
4605                     "200": {
4606                         "description": "",
4607                         "schema": { "$ref": "#/definitions/UpdateSchema" }
4608                     }
4609                 }
4610             }
4611         }
4612     },
4613     "parameters": {
4614         "interface-actuator" : {
4615             "in" : "query",
4616             "name" : "if",
4617             "type" : "string",
4618             "enum" : ["oic.if.a", "oic.if.baseline"]
4619         }
4620     },
4621     "definitions": {
4622         "RetrieveSchema" :
4623         {
4624             "properties": {
4625                 "automode": {
4626                     "description": "Auto mode status.",
4627                     "type": "integer",
4628                     "x-ocf-conversion": {
4629                         "x-from-ocf": [
4630                             "automode = ocf.automode",
4631                             "otherwise: automode = NotSupported(0xFF)"
4632                         ],
4633                         "x-ocf-alias": "oic.r.airflow",
4634                         "x-to-ocf": [
4635                             "if automode != NotSupported(0xFF)",
4636                             " ocf.automode = automode",
4637                             "else no mapping"

```

```

4638         ]
4639     },
4640 },
4641 "fanspeedlevel": {
4642     "description": "Fan speed level. 0 = off.",
4643     "type": "integer",
4644     "x-ocf-conversion": {
4645         "x-from-ocf": [
4646             "fanspeedlevel = speed"
4647         ],
4648         "x-ocf-alias": "oic.r.airflow",
4649         "x-to-ocf": [
4650             "speed = fanspeedlevel"
4651         ]
4652     }
4653 },
4654 "maxfanspeedlevel": {
4655     "description": "Max level allowed for fan speed",
4656     "type": "integer",
4657     "x-ocf-conversion": {
4658         "x-from-ocf": [
4659             "maxfanspeedlevel = range[1]",
4660             "otherwise: maxfanspeedlevel = 100"
4661         ],
4662         "x-ocf-alias": "oic.r.airflow",
4663         "x-to-ocf": [
4664             "range[0] = 0",
4665             "range[1] = maxfanspeedlevel"
4666         ]
4667     }
4668 },
4669 },
4670 "required": [
4671     "fanspeedlevel",
4672     "maxfanspeedlevel",
4673     "automode"
4674 ],
4675 "type": "object"
4676 }
4677
4678 ,
4679 "UpdateSchema" :
4680 {
4681     "properties": {
4682         "automode": {
4683             "description": "Auto mode status.",
4684             "type": "integer",
4685             "x-ocf-conversion": {
4686                 "x-from-ocf": [
4687                     "automode = ocf.automode",
4688                     "otherwise: automode = NotSupported(0xFF)"
4689                 ],
4690                 "x-ocf-alias": "oic.r.airflow",
4691                 "x-to-ocf": [
4692                     "if automode != NotSupported(0xFF)",
4693                     " ocf.automode = automode",
4694                     "else no mapping"
4695                 ]
4696             }
4697         },
4698         "fanspeedlevel": {
4699             "description": "Fan speed level. 0 = off.",
4700             "type": "integer",
4701             "x-ocf-conversion": {
4702                 "x-from-ocf": [
4703                     "fanspeedlevel = speed"
4704                 ],
4705                 "x-ocf-alias": "oic.r.airflow",
4706                 "x-to-ocf": [
4707                     "speed = fanspeedlevel"
4708                 ]

```

```

4709     }
4710   },
4711   "maxfanspeedlevel": {
4712     "description": "Max level allowed for fan speed",
4713     "type": "integer",
4714     "x-ocf-conversion": {
4715       "x-from-ocf": [
4716         "maxfanspeedlevel = range[1]",
4717         "otherwise: maxfanspeedlevel = 100"
4718       ],
4719       "x-ocf-alias": "oic.r.airflow",
4720       "x-to-ocf": [
4721         "range[0] = 0",
4722         "range[1] = maxfanspeedlevel"
4723       ]
4724     }
4725   },
4726   },
4727   "required": [
4728     "fanspeedlevel",
4729     "maxfanspeedlevel",
4730     "automode"
4731   ],
4732   "type": "object"
4733 }
4734
4735 }
4736 }
4737

```

4738 A.9.5 Property Definition

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

4739 A.9.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/FanSpeedLevelResURI		get	post		

4740 A.10 Heating Zone Mapping

4741 A.10.1 Introduction

4742 This API defines the mapping between an instance of an AllJoyn HeatingZone interface and an
 4743 OCF HeatingZoneCollection Resource.
 4744 Each element in the array of heating zones within the AllJoyn HeatingZone interface maps to an
 4745 instance of OCF HeatingZone, itself a link in an instance of an OCF HeatingZoneCollection.
 4746 The mapping defined in the schema describes the population of the OCF HeatingZone Resource
 4747 that constitutes the Resources that are contained in the collection.
 4748

4749 **A.10.2 Example URI**

4750 /HeatingZoneResURI

4751 **A.10.3 Resource Type**

4752 The resource type (rt) is defined as: ['oic.r.heatingzonecollection'].

4753 **A.10.4 Swagger2.0 Definition**

```
4754 {
4755   "swagger": "2.0",
4756   "info": {
4757     "title": "Heating Zone Mapping",
4758     "version": "OCFv1.0.0-20170317",
4759     "license": {
4760       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
4761       "x-description": "Redistribution and use in source and binary forms, with or without
4762 modification, are permitted provided that the following conditions are met:\n      1.
4763 Redistributions of source code must retain the above copyright notice, this list of conditions and
4764 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
4765 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
4766 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
4767 Connectivity Foundation, INC. \\"AS IS\\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
4768 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
4769 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
4770 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
4771 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
4772 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND
4773 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
4774 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
4775 OF SUCH DAMAGE.\n"
4776   },
4777 },
4778 "schemes": ["http"],
4779 "consumes": ["application/json"],
4780 "produces": ["application/json"],
4781 "paths": {
4782   "/HeatingZoneResURI" : {
4783     "get": {
4784       "description": "This API defines the mapping between an instance of an AllJoyn HeatingZone
4785 interface and an OCF HeatingZoneCollection Resource.\nEach element in the array of heating zones
4786 within the AllJoyn HeatingZone interface maps to an instance of OCF HeatingZone, itself a link in
4787 an instance of an OCF HeatingZoneCollection.\nThe mapping defined in the schema describes the
4788 population of the OCF HeatingZone Resource that constitutes the Resources that are contained in the
4789 collection.\n",
4790       "parameters": [
4791         {"$ref": "#/parameters/interface-sensor"}
4792       ],
4793       "responses": {
4794         "200": {
4795           "description": "",
4796           "x-example": {
4797             "rt": ["oic.r.heatingzonecollection"]
4798           },
4799           "schema": { "$ref": "#/definitions/RetrieveSchema" }
4800         }
4801       }
4802     }
4803   }
4804 },
4805 },
4806 },
4807 "parameters": {
4808   "interface-sensor" : {
4809     "in" : "query",
4810     "name" : "if",
4811     "type" : "string",
4812     "enum" : ["oic.if.s", "oic.if.baseline"]
4813   }
4814 },
4815 "definitions": {
```

```

4816     "RetrieveSchema" :
4817     {
4818         "properties": {
4819             "heatinglevels": {
4820                 "description": "Current heating levels for each zone.",
4821                 "items": {
4822                     "type": "integer"
4823                 },
4824                 "type": "array",
4825                 "x-ocf-conversion": {
4826                     "x-from-ocf": [
4827                         "for x=0;x<numlinks(oic.r.heatingzonecollection): heatinglevels[x] =
4828 ocf.heatinglevel"
4829                     ],
4830                     "x-ocf-alias": "oic.r.heatingzone",
4831                     "x-to-ocf": [
4832                         "Instance of oic.r.heatingzone per array item ",
4833                         "for x=0, x<sizeof(heatinglevels): ocf.heatinglevel = maxheatinglevels[x]"
4834                     ]
4835                 }
4836             },
4837             "maxheatinglevels": {
4838                 "description": "Max heating levels for each zone",
4839                 "items": {
4840                     "type": "integer"
4841                 },
4842                 "type": "array",
4843                 "x-ocf-conversion": {
4844                     "x-from-ocf": [
4845                         "for x=0;x<numlinks(oic.r.heatingzonecollection): maxheatinglevels[x] =
4846 ocf.maxheatinglevel"
4847                     ],
4848                     "x-ocf-alias": "oic.r.heatingzone",
4849                     "x-to-ocf": [
4850                         "Instance of oic.r.heatingzone per array item ",
4851                         "for x=0, x<sizeof(maxheatinglevels): ocf.maxheatinglevel = maxheatinglevels[x]"
4852                     ]
4853                 }
4854             },
4855             "numberofheatingzones": {
4856                 "description": "Number of heating zones.",
4857                 "type": "integer",
4858                 "x-ocf-conversion": {
4859                     "x-from-ocf": [
4860                         "numberofheatingzones = number of links in the collection"
4861                     ],
4862                     "x-ocf-alias": "oic.r.heatingzonecollection",
4863                     "x-to-ocf": [
4864                         "number of links in the collection = numberofheatingzones"
4865                     ]
4866                 }
4867             }
4868         },
4869         "required": [
4870             "numberofheatingzones",
4871             "maxheatinglevels",
4872             "heatinglevels"
4873         ],
4874         "type": "object"
4875     }
4876 }
4877 }
4878 }
4879

```

4880 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

4881 A.10.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/HeatingZoneResURI		get			

4882 A.11 HVAC Fan Mode Mapping

4883 A.11.1 Introduction

4884 This API defines the mapping between an instance of an AllJoyn HvacFanMode interface and an
4885 OCF Mode Resource.

4887 A.11.2 Example URI

4888 /HvacFanModeResURI

4889 A.11.3 Resource Type

4890 The resource type (rt) is defined as: ['oic.r.mode'].

4891 A.11.4 Swagger2.0 Definition

```

4892 {
4893   "swagger": "2.0",
4894   "info": {
4895     "title": "HVAC Fan Mode Mapping",
4896     "version": "OCFv1.0.0-20170317",
4897     "license": {
4898       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
4899       "x-description": "Redistribution and use in source and binary forms, with or without
4900 modification, are permitted provided that the following conditions are met:\n      1.
4901 Redistributions of source code must retain the above copyright notice, this list of conditions and
4902 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
4903 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
4904 other materials provided with the distribution.\n      THIS SOFTWARE IS PROVIDED BY THE Open
4905 Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
4906 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
4907 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
4908 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
4909 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

```

```

4910 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n          HOWEVER CAUSED AND
4911 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
4912 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
4913 OF SUCH DAMAGE.\n"
4914     }
4915 },
4916 "schemes": ["http"],
4917 "consumes": ["application/json"],
4918 "produces": ["application/json"],
4919 "paths": {
4920     "/HvacFanModeResURI" : {
4921         "get": {
4922             "description": "This API defines the mapping between an instance of an AllJoyn HvacFanMode
4923 interface and an OCF Mode Resource.\n",
4924             "parameters": [
4925                 {"$ref": "#/parameters/interface-actuator"}
4926             ],
4927             "responses": {
4928                 "200": {
4929                     "description": "",
4930                     "x-example": {
4931                         "rt": ["oic.r.mode"]
4932                     },
4933                     "schema": { "$ref": "#/definitions/RetrieveSchema" }
4934                 },
4935             },
4936         },
4937     },
4938 },
4939 "post": {
4940     "description": "",
4941     "parameters": [
4942         {"$ref": "#/parameters/interface-actuator"},
4943         {
4944             "name": "body",
4945             "in": "body",
4946             "required": true,
4947             "schema": { "$ref": "#/definitions/UpdateSchema" }
4948         }
4949     ],
4950     "responses": {
4951         "200": {
4952             "description": "",
4953             "schema": { "$ref": "#/definitions/UpdateSchema" }
4954         },
4955     },
4956 },
4957 },
4958 },
4959 "parameters": {
4960     "interface-actuator" : {
4961         "in" : "query",
4962         "name" : "if",
4963         "type" : "string",
4964         "enum" : ["oic.if.a", "oic.if.baseline"]
4965     },
4966 },
4967 "definitions": {
4968     "RetrieveSchema" :
4969     {
4970         "properties": {
4971             "mode": {
4972                 "description": "Current mode of device.",
4973                 "type": "integer",
4974                 "x-ocf-conversion": {
4975                     "x-from-ocf": [
4976                         "modearray = [Auto,Circulation,Continuous]",
4977                         "mode = indexOf modeArray[ocf.mode[0]]"
4978                     ],
4979                     "x-ocf-alias": "oic.r.mode",
4980                     "x-to-ocf": [

```

```

4981         "modearray = [Auto,Circulation,Continuous]",
4982         "ocf.mode[0] = modearray[mode]"
4983     ]
4984 }
4985 },
4986 "supportedmodes": {
4987     "description": "Array of supported modes",
4988     "items": {
4989         "type": "integer"
4990     },
4991     "type": "array",
4992     "x-ocf-conversion": {
4993         "x-from-ocf": [
4994             "modearray = [Auto,Circulation,Continuous]",
4995             "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
modearray[ocf.supportedmodes[x]]"
4996         ],
4997         "x-ocf-alias": "oic.r.mode",
4998         "x-to-ocf": [
4999             "modearray = [Auto,Circulation,Continuous]",
5000             "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
modearray[supportedmodes[x]]"
5001         ]
5002     }
5003 }
5004 }
5005 },
5006 },
5007 "required": [
5008     "mode",
5009     "supportedmodes"
5010 ],
5011 "type": "object"
5012 }
5013
5014 ,
5015 "UpdateSchema" :
5016 {
5017     "properties": {
5018         "mode": {
5019             "description": "Current mode of device.",
5020             "type": "integer",
5021             "x-ocf-conversion": {
5022                 "x-from-ocf": [
5023                     "modearray = [Auto,Circulation,Continuous]",
5024                     "mode = indexof modeArray[ocf.mode[0]]"
5025                 ],
5026                 "x-ocf-alias": "oic.r.mode",
5027                 "x-to-ocf": [
5028                     "modearray = [Auto,Circulation,Continuous]",
5029                     "ocf.mode[0] = modearray[mode]"
5030                 ]
5031             }
5032         },
5033         "supportedmodes": {
5034             "description": "Array of supported modes",
5035             "items": {
5036                 "type": "integer"
5037             },
5038             "type": "array",
5039             "x-ocf-conversion": {
5040                 "x-from-ocf": [
5041                     "modearray = [Auto,Circulation,Continuous]",
5042                     "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
modearray[ocf.supportedmodes[x]]"
5043                 ],
5044                 "x-ocf-alias": "oic.r.mode",
5045                 "x-to-ocf": [
5046                     "modearray = [Auto,Circulation,Continuous]",
5047                     "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
modearray[supportedmodes[x]]"
5048                 ]
5049             }
5050         }
5051     }

```

```

5052     }
5053   },
5054   "required": [
5055     "mode",
5056     "supportedmodes"
5057   ],
5058   "type": "object"
5059 }
5060
5061 }
5062 }
5063

```

5064 A.11.5 Property Definition

['AllJoyn'] Property name	OCF Resour ce	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.

5065 A.11.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/HvacFanModeResURI		get	post		

5066 A.12 On Off Mapping

5067 A.12.1 Introduction

5068 This API defines the mapping between an instance of an OCF Binary Switch Resource and the
5069 equivalent Interface set by AllJoyn. A discovered instance of a Binary Switch is always
5070 mapped to an Operation.OnOffStatus interface.
5071 A RETRIEVE on a Binary Switch maps to an action on an instance of an Operation.OnOffStatus
5072 Interface.

5073 An UPDATE on a Binary Switch maps to a method invocation on either Operation.OnControl or
5074 OffControl.

5075 value = true maps to Operation.OnControl
5076 value = false maps to Operation.OffControl

5077

5078 A.12.2 Example URI

5079 /OnOffResURI

5080 A.12.3 Resource Type

5081 The resource type (rt) is defined as: ['oic.r.switch.binary'].

5082 A.12.4 Swagger2.0 Definition

```

5083 {
5084   "swagger": "2.0",
5085   "info": {
5086     "title": "On Off Mapping",
5087     "version": "OCFv1.0.0-20170317",

```

```

5088     "license": {
5089         "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
5090         "x-description": "Redistribution and use in source and binary forms, with or without
5091 modification, are permitted provided that the following conditions are met:\n      1.
5092 Redistributions of source code must retain the above copyright notice, this list of conditions and
5093 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
5094 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
5095 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
5096 Connectivity Foundation, INC. \\"AS IS\\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
5097 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
5098 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n\n      IN NO EVENT SHALL THE Open Connectivity
5099 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
5100 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
5101 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n\n      HOWEVER CAUSED AND
5102 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
5103 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
5104 OF SUCH DAMAGE.\n"
5105     },
5106     "schemes": ["http"],
5107     "consumes": ["application/json"],
5108     "produces": ["application/json"],
5109     "paths": {
5110         "/OnOffResURI" : {
5111             "get": {
5112                 "description": "This API defines the mapping between an instance of an OCF Binary Switch
5113 Resource and the\nequivalent Interface set by AllJoyn. A discovered instance of a Binary Switch is
5114 always\mapped to an Operation.OnOffStatus interface.\nA RETRIEVE on a Binary Switch maps to an
5115 action on an instance of an Operation.OnOffStatus Interface.\nAn UPDATE on a Binary Switch maps to
5116 a method invocation on either Operation.OnControl or OffControl.\n value = true maps to
5117 Operation.OnControl\n value = false maps to Operation.OffControl\n",
5118                 "parameters": [
5119                     { "$ref": "#/parameters/interface-all" }
5120                 ],
5121                 "responses": {
5122                     "200": {
5123                         "description": "",
5124                         "x-example": {
5125                             "rt": ["oic.r.switch.binary"]
5126                         },
5127                         "schema": { "$ref": "#/definitions/RetrieveSchema" }
5128                     }
5129                 },
5130             },
5131             "post": {
5132                 "description": "",
5133                 "parameters": [
5134                     { "$ref": "#/parameters/interface-actuator" },
5135                     {
5136                         "name": "body",
5137                         "in": "body",
5138                         "required": true,
5139                         "schema": { "$ref": "#/definitions/UpdateSchema" }
5140                     }
5141                 ],
5142                 "responses": {
5143                     "200": {
5144                         "description": "",
5145                         "schema": { "$ref": "#/definitions/UpdateSchema" }
5146                     }
5147                 },
5148             },
5149         },
5150     },
5151     "parameters": {
5152         "interface-actuator" : {
5153             "in" : "query",
5154             "name" : "if",
5155             "type" : "string",

```

```

5159         "enum" : ["oic.if.a", "oic.if.baseline"]
5160     },
5161     "interface-all" : {
5162         "in" : "query",
5163         "name" : "if",
5164         "type" : "string",
5165         "enum" : ["oic.if.s", "oic.if.a", "oic.if.baseline"]
5166     }
5167 },
5168 "definitions": {
5169     "RetrieveSchema" :
5170     {
5171         "properties": {
5172             "onoff": {
5173                 "description": "On/Off status of the device",
5174                 "type": "boolean",
5175                 "x-ocf-conversion": {
5176                     "x-from-ocf": [
5177                         "onoff = value"
5178                     ],
5179                     "x-ocf-alias": "oic.r.switch.binary",
5180                     "x-to-ocf": [
5181                         "value = onoff"
5182                     ]
5183                 }
5184             },
5185             "required": [
5186                 "onoff"
5187             ],
5188             "type": "object"
5189         }
5190     },
5191     "UpdateSchema" :
5192     {
5193         "properties": {
5194             "switchon": {
5195                 "description": "Turn on the device",
5196                 "format": "method",
5197                 "type": "string",
5198                 "x-ocf-conversion": {
5199                     "x-ocf-alias": "oic.r.switch.binary",
5200                     "x-to-ocf": [
5201                         "value = true"
5202                     ]
5203                 }
5204             },
5205             "type": "object"
5206         }
5207     },
5208     "type": "object"
5209 }
5210 }
5211 }
5212 }
5213

```

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

A.12.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/OnOffResURI		get	post		

A.13 Oven Cycle Phase Mapping

A.13.1 Introduction

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

A.13.2 Example URI

/OvenCyclePhaseResURI

A.13.3 Resource Type

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

A.13.4 Swagger2.0 Definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Oven Cycle Phase 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      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      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": {
    "/OvenCyclePhaseResURI" : {
      "get": {
        "description": "This API defines the mapping between an instance of an AllJoyn
OvenCyclePhase interface and the OCF OperationalState Resource.\nOvenCyclePhase cyclephase Property
pre-defines values 0x00-0x7F, 0x80-0xFF is for vendor specific values. The mapping defined herein
covers only Spec defined values.\nAny vendor defined value shall be represented in OCF using the
x.<organization> syntax for a vendor defined Property.\nThe AllJoyn interface also supports a
Method, GetVendorPhasesDescription; this is handled in OCF using an instance of oic.r.actuator
within an oic.r.action collection.\nPlease see Section 8 of the Mapping Specification for
specifics.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-sensor"}
        ],
        "responses": {
          "200": {
            "description": "",

```

```

5277         "x-example":
5278             {
5279                 "rt":      ["oic.r.operationalstate"]
5280             }
5281         ,
5282         "schema": { "$ref": "#/definitions/RetrieveSchema" }
5283     }
5284 }
5285 }
5286 }
5287 },
5288 "parameters": {
5289     "interface-sensor" : {
5290         "in" : "query",
5291         "name" : "if",
5292         "type" : "string",
5293         "enum" : ["oic.if.s", "oic.if.baseline"]
5294     }
5295 },
5296 "definitions": {
5297     "RetrieveSchema" :
5298     {
5299         "properties": {
5300             "cyclephase": {
5301                 "description": "Current phase of the operational cycle",
5302                 "type": "integer",
5303                 "x-ocf-conversion": {
5304                     "x-from-ocf": [
5305                         "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
5306                         "cyclephase = indexof statearray[currentmachinestate[0]]"
5307                     ],
5308                     "x-ocf-alias": "oic.r.operationalstate",
5309                     "x-to-ocf": [
5310                         "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
5311                         "currentmachinestate = phasearray[cyclephase]"
5312                     ]
5313                 }
5314             },
5315             "getvendorphasesdescription": {
5316                 "description": "Get cycle phases description",
5317                 "x-ocf-conversion": {
5318                     "x-ocf-alias": "oic.r.action"
5319                 },
5320                 "x-ocf-type": "method"
5321             },
5322             "supportedcyclephases": {
5323                 "description": "Array of cycle phases supported by the Appliance.",
5324                 "items": {
5325                     "type": "integer"
5326                 },
5327                 "type": "array",
5328                 "x-ocf-conversion": {
5329                     "x-from-ocf": [
5330                         "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
5331                         "for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof
5332 phasearray[machinestates[x]]"
5333                     ],
5334                     "x-ocf-alias": "oic.r.operationalstate",
5335                     "x-to-ocf": [
5336                         "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
5337                         "for x=0, x < sizeof(supportedcyclephases): machinestates[x] =
5338 phasearray[supportedcyclephases[x]]"
5339                     ]
5340                 }
5341             }
5342         },
5343         "required": [
5344             "cyclephase",
5345             "supportedcyclephases"
5346         ],
5347         "type": "object"

```

5348 }
5349
5350 }
5351 }
5352

5353 **A.13.5 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

5354 **A.13.6 CRUDN behavior**

Resource	Create	Read	Update	Delete	Notify
/OvenCyclePhaseResURI		get			

5355 **A.14 Target Humidity Mapping**

5356 **A.14.1 Introduction**

5357 This API defines the mapping between an instance of an AllJoyn TargetHumidity Interface and the
5358 OCF Resource Equivalent.
5359 A POST on a Humidity Sensor maps to an action on an instance of an Environment.TargetHumidity
5360 Interface.
5361

5362 **A.14.2 Example URI**

5363 /TargetHumidityResURI

5364 **A.14.3 Resource Type**

5365 The resource type (rt) is defined as: ['oic.r.humidity', 'oic.r.selectablelevels'].

A.14.4 Swagger2.0 Definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Target Humidity 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      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      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": {
    "/TargetHumidityResURI" : {
      "get": {
        "description": "This API defines the mapping between an instance of an AllJoyn
TargetHumidity Interface and the OCF Resource Equivalent.\nA POST on a Humidity Sensor maps to an
action on an instance of an Environment.TargetHumidity Interface.\n",
        "parameters": [
          { "$ref": "#/parameters/interface-actuator" }
        ],
        "responses": {
          "200": {
            "description": "",
            "x-example": {
              "rt": ["oic.r.humidity", "oic.r.selectablelevels"]
            },
            "schema": { "$ref": "#/definitions/RetrieveSchema" }
          }
        }
      },
      "post": {
        "description": "",
        "parameters": [
          { "$ref": "#/parameters/interface-actuator" },
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": { "$ref": "#/definitions/UpdateSchema" }
          }
        ],
        "responses": {
          "200": {
            "description": "",
            "x-example": {
              "rt": ["oic.r.humidity", "oic.r.selectablelevels"]
            },
            "schema": { "$ref": "#/definitions/UpdateSchema" }
          }
        }
      }
    }
  }
}
```

```

5436     }
5437   }
5438 }
5439 },
5440 "parameters": {
5441   "interface-actuator" : {
5442     "in" : "query",
5443     "name" : "if",
5444     "type" : "string",
5445     "enum" : ["oic.if.a", "oic.if.baseline"]
5446   }
5447 },
5448 "definitions": {
5449   "RetrieveSchema" :
5450     {
5451     "properties": {
5452       "maxvalue": {
5453         "type": "number",
5454         "x-ocf-conversion": {
5455           "x-from-ocf": [
5456             "maxvalue = range[1]",
5457             "otherwise: maxvalue = 100"
5458           ],
5459           "x-ocf-alias": "oic.r.humidity",
5460           "x-to-ocf": [
5461             "range[1] = maxvalue"
5462           ]
5463         }
5464       },
5465       "minvalue": {
5466         "type": "number",
5467         "x-ocf-conversion": {
5468           "x-from-ocf": [
5469             "minvalue = range[0]",
5470             "otherwise: minvalue = 0"
5471           ],
5472           "x-ocf-alias": "oic.r.humidity",
5473           "x-to-ocf": [
5474             "range[0] = minvalue"
5475           ]
5476         }
5477       },
5478       "selectablehumiditylevels": {
5479         "items": {
5480           "type": "number"
5481         },
5482         "type": "array",
5483         "x-ocf-conversion": {
5484           "x-from-ocf": [
5485             "selectablehumiditylevels[] = availablelevels[]"
5486           ],
5487           "x-ocf-alias": "oic.r.selectablelevels",
5488           "x-to-ocf": [
5489             "availablelevels[] = selectablehumiditylevels[]"
5490           ]
5491         }
5492       },
5493       "stepvalue": {
5494         "type": "number",
5495         "x-ocf-conversion": {
5496           "x-from-ocf": [
5497             "stepvalue = step",
5498             "otherwise: stepvalue = 1"
5499           ],
5500           "x-ocf-alias": "oic.r.humidity",
5501           "x-to-ocf": [
5502             "step = stepvalue"
5503           ]
5504         }
5505       },
5506       "targetvalue": {

```

```

5507         "description": "Measured value",
5508         "type": "number",
5509         "x-ocf-conversion": {
5510             "x-from-ocf": [
5511                 "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
5512                 "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
5513             ],
5514             "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
5515             "x-to-ocf": [
5516                 "if minvalue != maxvalue, ocf.desiredhumidity = targetvalue;ocf.targetlevel =
5517 selectablehumiditylevels[0].",
5518                 "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
5519             ]
5520         }
5521     },
5522 },
5523 "required": [
5524     "targetvalue",
5525     "minvalue",
5526     "maxvalue",
5527     "stepvalue",
5528     "selectablehumiditylevels"
5529 ],
5530 "type": "object"
5531 }
5532
5533 ,
5534 "UpdateSchema" :
5535 {
5536     "properties": {
5537         "maxvalue": {
5538             "type": "number",
5539             "x-ocf-conversion": {
5540                 "x-from-ocf": [
5541                     "maxvalue = range[1]",
5542                     "otherwise: maxvalue = 100"
5543                 ],
5544                 "x-ocf-alias": "oic.r.humidity",
5545                 "x-to-ocf": [
5546                     "range[1] = maxvalue"
5547                 ]
5548             }
5549         },
5550         "minvalue": {
5551             "type": "number",
5552             "x-ocf-conversion": {
5553                 "x-from-ocf": [
5554                     "minvalue = range[0]",
5555                     "otherwise: minvalue = 0"
5556                 ],
5557                 "x-ocf-alias": "oic.r.humidity",
5558                 "x-to-ocf": [
5559                     "range[0] = minvalue"
5560                 ]
5561             }
5562         },
5563         "selectablehumiditylevels": {
5564             "items": {
5565                 "type": "number"
5566             },
5567             "type": "array",
5568             "x-ocf-conversion": {
5569                 "x-from-ocf": [
5570                     "selectablehumiditylevels[] = availablelevels[]"
5571                 ],
5572                 "x-ocf-alias": "oic.r.selectablelevels",
5573                 "x-to-ocf": [
5574                     "availablelevels[] = selectablehumiditylevels[]"
5575                 ]
5576             }
5577         }
5578     },

```

```

5578     "stepvalue": {
5579         "type": "number",
5580         "x-ocf-conversion": {
5581             "x-from-ocf": [
5582                 "stepvalue = step",
5583                 "otherwise: stepvalue = 1"
5584             ],
5585             "x-ocf-alias": "oic.r.humidity",
5586             "x-to-ocf": [
5587                 "step = stepvalue"
5588             ]
5589         }
5590     },
5591     "targetvalue": {
5592         "description": "Measured value",
5593         "type": "number",
5594         "x-ocf-conversion": {
5595             "x-from-ocf": [
5596                 "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
5597                 "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
5598             ],
5599             "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
5600             "x-to-ocf": [
5601                 "if minvalue != maxvalue, ocf.desiredhumidity = targetvalue;ocf.targetlevel =
5602 selectablehumiditylevels[0].",
5603                 "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
5604             ]
5605         }
5606     },
5607 ],
5608 "required": [
5609     "targetvalue",
5610     "minvalue",
5611     "maxvalue",
5612     "stepvalue",
5613     "selectablehumiditylevels"
5614 ],
5615 "type": "object"
5616 }
5617
5618 }
5619 }
5620

```

A.14.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Descrip tion
stepvalue	oic.r.humidity	step = stepvalue	stepvalue = stepotherwise: stepvalue = 1	
targetvalue	oic.r.humidity,oic.r.sele ctablelevels	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	

A.14.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/TargetHumidityResURI		get	post		

A.15 Target Temperature Mapping

A.15.1 Introduction

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

A.15.2 Example URI

/TargetTemperatureResURI

A.15.3 Resource Type

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

A.15.4 Swagger2.0 Definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Target Temperature 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      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      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": {
    "/TargetTemperatureResURI" : {
      "get": {
        "description": "This API defines the mapping between an instance of an OCF Temperature
which exposes only a sensor interface\nand the AllJoyn Current Temperature interface.\nA RETRIEVE
on a Temperature Sensor maps to an action on an instance of an Environment.CurrentTemperature
Interface.\n",
        "parameters": [
```



```

5671         {"$ref": "#/parameters/interface-actuator"}
5672     ],
5673     "responses": {
5674         "200": {
5675             "description": "",
5676             "x-example": {
5677                 "rt": ["oic.r.temperature"]
5678             }
5679         },
5680         "schema": { "$ref": "#/definitions/RetrieveSchema" }
5681     }
5682 },
5683 },
5684 },
5685 "post": {
5686     "description": "",
5687     "parameters": [
5688         {"$ref": "#/parameters/interface-actuator"},
5689         {
5690             "name": "body",
5691             "in": "body",
5692             "required": true,
5693             "schema": { "$ref": "#/definitions/UpdateSchema" }
5694         }
5695     ],
5696     "responses": {
5697         "200": {
5698             "description": "",
5699             "schema": { "$ref": "#/definitions/UpdateSchema" }
5700         }
5701     }
5702 }
5703 },
5704 },
5705 "parameters": {
5706     "interface-actuator" : {
5707         "in" : "query",
5708         "name" : "if",
5709         "type" : "string",
5710         "enum" : ["oic.if.a", "oic.if.baseline"]
5711     }
5712 },
5713 "definitions": {
5714     "RetrieveSchema" :
5715     {
5716         "properties": {
5717             "maxvalue": {
5718                 "type": "number",
5719                 "x-ocf-conversion": {
5720                     "x-from-ocf": [
5721                         "maxvalue = range[1]",
5722                         "otherwise: maxvalue = MAXINT"
5723                     ],
5724                     "x-ocf-alias": "oic.r.temperature",
5725                     "x-to-ocf": [
5726                         "range[1] = maxvalue"
5727                     ]
5728                 }
5729             },
5730             "minvalue": {
5731                 "type": "number",
5732                 "x-ocf-conversion": {
5733                     "x-from-ocf": [
5734                         "minvalue = range[0]",
5735                         "otherwise: minvalue = -MAXINT"
5736                     ],
5737                     "x-ocf-alias": "oic.r.temperature",
5738                     "x-to-ocf": [
5739                         "range[0] = minvalue"
5740                     ]
5741                 }

```

```

5742     },
5743     "step": {
5744         "type": "number",
5745         "x-ocf-conversion": {
5746             "x-from-ocf": [
5747                 "step = ocf.step",
5748                 "otherwise: step = undefined (0x00)"
5749             ],
5750             "x-ocf-alias": "oic.r.temperature",
5751             "x-to-ocf": [
5752                 "ocf.step = step"
5753             ]
5754         }
5755     },
5756     "targetvalue": {
5757         "description": "Measured value",
5758         "type": "number",
5759         "x-ocf-conversion": {
5760             "x-from-ocf": {
5761                 "oneOf": [
5762                     {
5763                         "properties": {
5764                             "enum": [
5765                                 "C"
5766                             ],
5767                             "units": "string"
5768                         },
5769                         "x-from-ocf": [
5770                             "targetvalue = temperature"
5771                         ]
5772                     },
5773                     {
5774                         "properties": {
5775                             "enum": [
5776                                 "F"
5777                             ],
5778                             "units": "string"
5779                         },
5780                         "x-from-ocf": [
5781                             "targetvalue = (temperature-32)*5/9"
5782                         ]
5783                     },
5784                     {
5785                         "properties": {
5786                             "enum": [
5787                                 "K"
5788                             ],
5789                             "units": "string"
5790                         },
5791                         "x-from-ocf": [
5792                             "targetvalue = temperature-273.15"
5793                         ]
5794                     }
5795                 ]
5796             },
5797             "x-ocf-alias": "oic.r.temperature",
5798             "x-to-ocf": [
5799                 "temperature = targetvalue",
5800                 "units = C"
5801             ]
5802         }
5803     }
5804 },
5805 "required": [
5806     "targetvalue",
5807     "minvalue",
5808     "maxvalue",
5809     "step"
5810 ],
5811 "type": "object"
5812 }

```

```

5813
5814
5815     "UpdateSchema" :
5816     {
5817         "properties": {
5818             "maxvalue": {
5819                 "type": "number",
5820                 "x-ocf-conversion": {
5821                     "x-from-ocf": [
5822                         "maxvalue = range[1]",
5823                         "otherwise: maxvalue = MAXINT"
5824                     ],
5825                     "x-ocf-alias": "oic.r.temperature",
5826                     "x-to-ocf": [
5827                         "range[1] = maxvalue"
5828                     ]
5829                 }
5830             },
5831             "minvalue": {
5832                 "type": "number",
5833                 "x-ocf-conversion": {
5834                     "x-from-ocf": [
5835                         "minvalue = range[0]",
5836                         "otherwise: minvalue = -MAXINT"
5837                     ],
5838                     "x-ocf-alias": "oic.r.temperature",
5839                     "x-to-ocf": [
5840                         "range[0] = minvalue"
5841                     ]
5842                 }
5843             },
5844             "step": {
5845                 "type": "number",
5846                 "x-ocf-conversion": {
5847                     "x-from-ocf": [
5848                         "step = ocf.step",
5849                         "otherwise: step = undefined (0x00)"
5850                     ],
5851                     "x-ocf-alias": "oic.r.temperature",
5852                     "x-to-ocf": [
5853                         "ocf.step = step"
5854                     ]
5855                 }
5856             },
5857             "targetvalue": {
5858                 "description": "Measured value",
5859                 "type": "number",
5860                 "x-ocf-conversion": {
5861                     "x-from-ocf": {
5862                         "oneOf": [
5863                             {
5864                                 "properties": {
5865                                     "enum": [
5866                                         "C"
5867                                     ],
5868                                     "units": "string"
5869                                 },
5870                                 "x-from-ocf": [
5871                                     "targetvalue = temperature"
5872                                 ]
5873                             },
5874                             {
5875                                 "properties": {
5876                                     "enum": [
5877                                         "F"
5878                                     ],
5879                                     "units": "string"
5880                                 },
5881                                 "x-from-ocf": [
5882                                     "targetvalue = (temperature-32)*5/9"
5883                                 ]

```

```

5884         },
5885         {
5886             "properties": {
5887                 "enum": [
5888                     "K"
5889                 ],
5890                 "units": "string"
5891             },
5892             "x-from-ocf": [
5893                 "targetvalue = temperature-273.15"
5894             ]
5895         }
5896     ],
5897 },
5898 "x-ocf-alias": "oic.r.temperature",
5899 "x-to-ocf": [
5900     "temperature = targetvalue",
5901     "units = C"
5902 ]
5903 }
5904 }
5905 },
5906 "required": [
5907     "targetvalue",
5908     "minvalue",
5909     "maxvalue",
5910     "step"
5911 ],
5912 "type": "object"
5913 }
5914 }
5915 }
5916 }
5917

```

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	

A.15.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/TargetTemperatureResURI		get	post		