

# OCF Resource to Zigbee Cluster Mapping Specification

VERSION 2.0.5 | September 2019



**OPEN** CONNECTIVITY  
FOUNDATION™

CONTACT [admin@openconnectivity.org](mailto:admin@openconnectivity.org)

Copyright Open Connectivity Foundation, Inc. © 2019.  
All Rights Reserved.

3 Legal Disclaimer

4

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

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

18 Copyright © 2019 Open Connectivity Foundation, Inc. All rights reserved.

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

20

## CONTENTS

22	1	Scope .....	1
23	2	Normative references .....	1
24	3	Terms, definitions symbols and abbreviations .....	1
25	4	Document conventions and organization .....	2
26	4.1	Conventions .....	2
27	4.2	Notation.....	2
28	5	Theory of Operation .....	3
29	5.1	Interworking Approach.....	3
30	5.2	Mapping Syntax.....	3
31	5.2.1	Introduction .....	3
32	5.2.2	General.....	3
33	5.2.3	Value Assignment .....	3
34	5.2.4	Property Naming .....	3
35	5.2.5	Range .....	3
36	5.2.6	Arrays .....	3
37	5.2.7	Default Mapping .....	3
38	5.2.8	Conditional Mapping.....	3
39	5.2.9	Method Invocation.....	4
40	6	Device Type Mapping.....	4
41	6.1	Introduction .....	4
42	6.2	Zigbee Device Types to OCF Device Types.....	4
43	7	Resource to ZigBee Cluster Equivalence.....	5
44	7.1	Introduction .....	5
45	7.2	Zigbee Clusters to OCF Resources.....	5
46	7.2.1	Introduction .....	5
47	7.2.2	On/off.....	6
48	7.2.3	Level Control.....	6
49	7.2.4	Color Control.....	6
50	7.2.5	Thermostat.....	7
51	7.2.6	Window Covering .....	7
52	7.2.7	Temperature Measurement .....	8
53	7.2.8	Occupancy Sensing.....	8
54	7.2.9	IAS Zone.....	8
55	8	Detailed Mapping APIs .....	9
56	8.1.1	Introduction .....	9
57	8.2	Color Control Cluster - Color Space - Control .....	9
58	8.2.1	Derived model .....	9
59	8.2.2	Property definition .....	9
60	8.2.3	Derived model definition.....	10
61	8.3	Color Control Cluster - Color Space - Information .....	10
62	8.3.1	Derived model.....	10
63	8.3.2	Property definition .....	10

64	8.3.3	Derived model definition.....	11
65	8.4	Color Control Cluster - Color Temperature - Information.....	12
66	8.4.1	Derived model.....	12
67	8.4.2	Property definition.....	12
68	8.4.3	Derived model definition.....	12
69	8.5	Color Control Cluster - Color Temperature - Information.....	13
70	8.5.1	Derived model.....	13
71	8.5.2	Property definition.....	13
72	8.5.3	Derived model definition.....	13
73	8.6	Color Control Cluster - Hue and Saturation - Control.....	14
74	8.6.1	Derived model.....	14
75	8.6.2	Property definition.....	14
76	8.6.3	Derived model definition.....	15
77	8.7	Color Control Cluster - Hue and Saturation - Information.....	16
78	8.7.1	Derived model.....	16
79	8.7.2	Property definition.....	16
80	8.7.3	Derived model definition.....	16
81	8.8	IAS Zone Cluster - Control.....	17
82	8.8.1	Derived model.....	17
83	8.8.2	Property definition.....	17
84	8.8.3	Derived model definition.....	17
85	8.9	IAS Zone Cluster - Information.....	18
86	8.9.1	Derived model.....	18
87	8.9.2	Property definition.....	18
88	8.9.3	Derived model definition.....	24
89	8.10	Level Control Cluster - Control.....	27
90	8.10.1	Derived model.....	27
91	8.10.2	Property definition.....	27
92	8.10.3	Derived model definition.....	28
93	8.11	Level Control Cluster - Information.....	28
94	8.11.1	Derived model.....	28
95	8.11.2	Property definition.....	28
96	8.11.3	Derived model definition.....	29
97	8.12	Occupancy Sensing Cluster - Information.....	29
98	8.12.1	Derived model.....	29
99	8.12.2	Property definition.....	29
100	8.12.3	Derived model definition.....	30
101	8.13	On/Off Cluster - Control.....	30
102	8.13.1	Derived model.....	30
103	8.13.2	Property definition.....	30
104	8.13.3	Derived model definition.....	31
105	8.14	On/off Cluster - Information.....	32
106	8.14.1	Derived model.....	32
107	8.14.2	Property definition.....	32

108	8.14.3	Derived model definition.....	32
109	8.15	Temperature Measurement Cluster - Information.....	33
110	8.15.1	Derived model.....	33
111	8.15.2	Property definition.....	33
112	8.15.3	Derived model definition.....	33
113	8.16	Thermostat Cluster - Cool - Control.....	34
114	8.16.1	Derived model.....	34
115	8.16.2	Property definition.....	34
116	8.16.3	Derived model definition.....	35
117	8.17	Thermostat Cluster - Current Temperature - Information.....	35
118	8.17.1	Derived model.....	35
119	8.17.2	Property definition.....	36
120	8.17.3	Derived model definition.....	36
121	8.18	Thermostat Cluster - Heat - Control.....	36
122	8.18.1	Derived model.....	36
123	8.18.2	Property definition.....	36
124	8.18.3	Derived model definition.....	37
125	8.19	Window Covering Cluster - Configuration - Control.....	37
126	8.19.1	Derived model.....	37
127	8.19.2	Property definition.....	38
128	8.19.3	Derived model definition.....	39
129	8.20	Window Covering Cluster - Configuration - Information.....	40
130	8.20.1	Derived model.....	40
131	8.20.2	Property definition.....	40
132	8.20.3	Derived model definition.....	44
133	8.21	Window Covering Cluster - Lift Percentage - Control.....	46
134	8.21.1	Derived model.....	46
135	8.21.2	Property definition.....	46
136	8.21.3	Derived model definition.....	46
137	8.22	Window Covering Cluster - Lift Percentage - Information.....	47
138	8.22.1	Derived model.....	47
139	8.22.2	Property definition.....	47
140	8.22.3	Derived model definition.....	47
141	8.23	Window Covering Cluster - Lift Position - Control.....	48
142	8.23.1	Derived model.....	48
143	8.23.2	Property definition.....	48
144	8.23.3	Derived model definition.....	48
145	8.24	Window Covering Cluster - Lift Position - Information.....	49
146	8.24.1	Derived model.....	49
147	8.24.2	Property definition.....	49
148	8.24.3	Derived model definition.....	50
149	8.25	Window Covering Cluster - Tilt Percentage - Control.....	50
150	8.25.1	Derived model.....	50
151	8.25.2	Property definition.....	51

152	8.25.3	Derived model definition.....	51
153	8.26	Window Covering Cluster - Tilt Percentage - Information .....	52
154	8.26.1	Derived model .....	52
155	8.26.2	Property definition .....	52
156	8.26.3	Derived model definition.....	52
157	8.27	Window Covering Cluster - Tilt Position - Control .....	52
158	8.27.1	Derived model .....	52
159	8.27.2	Property definition .....	53
160	8.27.3	Derived model definition.....	53
161	8.28	Window Covering Cluster - Tilt Position - Information .....	53
162	8.28.1	Derived model .....	53
163	8.28.2	Property definition .....	54
164	8.28.3	Derived model definition.....	54
165			
166			

167

## Figures

168 **No table of figures entries found.**

## Tables

170	Table 1 – Zigbee to OCF Device Type Mapping .....	4
171	Table 2 – Zigbee Server Cluster to OCF Resource Type Mapping .....	5
172	Table 3 – The Property mapping for "zcl.colorcontrol_csc.control.movetocolor" .....	9
173	Table 4 – The Properties of "zcl.colorcontrol_csc.control.movetocolor" .....	9
174	Table 5 – The Property mapping for "zcl.colorcontrol_csc.info" .....	10
175	Table 6 – The Properties of "zcl.colorcontrol_csc.info" .....	11
176	Table 7 – The Property mapping for "zcl.colorcontrol_ct.control.movetocolortemperature" ....	12
177	Table 8 – The Properties of "zcl.colorcontrol_ct.control.movetocolortemperature" .....	12
178	Table 9 – The Property mapping for "zcl.colorcontrol_ct.info" .....	13
179	Table 10 – The Properties of "zcl.colorcontrol_ct.info" .....	13
180	Table 11 – The Property mapping for	
181	"zcl.colorcontrol_hs.control.movetohueandsaturation" .....	14
182	Table 12 – The Properties of "zcl.colorcontrol_hs.control.movetohueandsaturation" .....	15
183	Table 13 – The Property mapping for "zcl.colorcontrol_hs.info" .....	16
184	Table 14 – The Properties of "zcl.colorcontrol_hs.info" .....	16
185	Table 15 – The Property mapping for "zcl.iaszone.control" .....	17
186	Table 16 – The Properties of "zcl.iaszone.control" .....	17
187	Table 17 – The Property mapping for "zcl.iaszone.info" .....	18
188	Table 18 – The Properties of "zcl.iaszone.info" .....	22
189	Table 19 – The Property mapping for "zcl.levelcontrol.control.moveto" .....	28
190	Table 20 – The Properties of "zcl.levelcontrol.control.moveto" .....	28
191	Table 21 – The Property mapping for "zcl.levelcontrol.info" .....	29
192	Table 22 – The Properties of "zcl.levelcontrol.info" .....	29
193	Table 23 – The Property mapping for "zcl.occupancysensing.info" .....	29
194	Table 24 – The Properties of "zcl.occupancysensing.info" .....	30
195	Table 25 – The Property mapping for "zcl.onoff.control.off" .....	31
196	Table 26 – The Properties of "zcl.onoff.control.off" .....	31
197	Table 27 – The Property mapping for "zcl.onoff.control.on" .....	31
198	Table 28 – The Properties of "zcl.onoff.control.on" .....	31
199	Table 29 – The Property mapping for "zcl.onoff" .....	32
200	Table 30 – The Properties of "zcl.onoff" .....	32
201	Table 31 – The Property mapping for "zcl.temperaturemeasurement.info" .....	33
202	Table 32 – The Properties of "zcl.temperaturemeasurement.info" .....	33
203	Table 33 – The Property mapping for "zcl.thermostat_cool.control.setpointraiselower" .....	35
204	Table 34 – The Properties of "zcl.thermostat_cool.control.setpointraiselower" .....	35
205	Table 35 – The Property mapping for "zcl.thermostat_currenttemperature.info" .....	36
206	Table 36 – The Properties of "zcl.thermostat_currenttemperature.info" .....	36
207	Table 37 – The Property mapping for "zcl.thermostat_heat.control.setpointraiseLower" .....	37



208	Table 38 – The Properties of "zcl.thermostat_heat.control.setpointraiseLower".	37
209	Table 39 – The Property mapping for "zcl.windowcovering_conf.control".	38
210	Table 40 – The Properties of "zcl.windowcovering_conf.control".	38
211	Table 41 – The Property mapping for "zcl.windowcovering_conf.info".	40
212	Table 42 – The Properties of "zcl.windowcovering_conf.info".	42
213	Table 43 – The Property mapping for	
214	"/zcl.windowcovering_liftpercentage.control.gotoliftpercentage".	46
215	Table 44 – The Properties of	
216	"/zcl.windowcovering_liftpercentage.control.gotoliftpercentage".	46
217	Table 45 – The Property mapping for "zcl.windowcovering_liftpercentage.info".	47
218	Table 46 – The Properties of "zcl.windowcovering_liftpercentage.info".	47
219	Table 47 – The Property mapping for	
220	"zcl.windowcovering_liftposition.control.gotoliftvalue".	48
221	Table 48 – The Properties of "zcl.windowcovering_liftposition.control.gotoliftvalue".	48
222	Table 49 – The Property mapping for "/zcl.windowcovering_liftposition.info".	49
223	Table 50 – The Properties of "/zcl.windowcovering_liftposition.info".	49
224	Table 51 – The Property mapping for	
225	"zcl.windowcovering_tiltpercentage.control.gototiltpercentage".	51
226	Table 52 – The Properties of	
227	"zcl.windowcovering_tiltpercentage.control.gototiltpercentage".	51
228	Table 53 – The Property mapping for "zcl.windowcovering_tiltpercentage.info".	52
229	Table 54 – The Properties of "zcl.windowcovering_tiltpercentage.info".	52
230	Table 55 – The Property mapping for	
231	"zcl.windowcovering_tiltposition.control.gototiltvalue".	53
232	Table 56 – The Properties of "zcl.windowcovering_tiltposition.control.gototiltvalue".	53
233	Table 57 – The Property mapping for "zcl.windowcovering_tiltposition.info".	54
234	Table 58 – The Properties of "zcl.windowcovering_tiltposition.info".	54
235		
236		

237 **1 Scope**

238 This document provides detailed mapping information between Zigbee defined Clusters and OCF  
239 defined Resources.

240 **2 Normative references**

241 The following documents are referred to in the text in such a way that some or all of their content  
242 constitutes requirements of this document. For dated references, only the edition cited applies.  
243 For undated references, the latest edition of the referenced document (including any amendments)  
244 applies.

245 ISO/IEC 30118-1:2018 Information technology -- Open Connectivity Foundation (OCF)  
246 Specification -- Part 1: Core specification  
247 <https://www.iso.org/standard/53238.html>  
248 Latest version available at: [https://openconnectivity.org/specs/OCF\\_Core\\_Specification.pdf](https://openconnectivity.org/specs/OCF_Core_Specification.pdf)

249 ISO/IEC 30118-2:2019, Information technology – Open Connectivity Foundation (OCF)  
250 Specification – Part 2: Security specification  
251 <https://www.iso.org/standard/74239.html>  
252 Latest version available at: [https://openconnectivity.org/specs/OCF\\_Security\\_Specification.pdf](https://openconnectivity.org/specs/OCF_Security_Specification.pdf)

253 ISO/IEC 30118-3:2019, Information technology – Open Connectivity Foundation (OCF)  
254 Specification – Part 3: Bridging specification  
255 <https://www.iso.org/standard/74240.html>  
256 Latest version available at: [https://openconnectivity.org/specs/OCF\\_Bridging\\_Specification.pdf](https://openconnectivity.org/specs/OCF_Bridging_Specification.pdf)

257 ISO/IEC 30118-4:2019, Information technology – Open Connectivity Foundation  
(OCF) 258 Specification – Part 4: Resource Type specification  
259 <https://www.iso.org/standard/74241.html>  
260 Latest version available at:  
261 [https://openconnectivity.org/specs/OCF\\_Resource\\_Type\\_Specification.pdf](https://openconnectivity.org/specs/OCF_Resource_Type_Specification.pdf)

262 ISO/IEC 30118-5:2019, Information technology – Open Connectivity Foundation (OCF)  
263 Specification – Part 5: Device specification  
264 <https://www.iso.org/standard/74242.html>  
265 Latest version available at: [https://openconnectivity.org/specs/OCF\\_Device\\_Specification.pdf](https://openconnectivity.org/specs/OCF_Device_Specification.pdf)

266 Derived Models for Interoperability between IoT Ecosystems, Stevens & Merriam, March 2016  
267 [https://www.iab.org/wp-content/IAB-uploads/2016/03/OCF-Derived-Models-for-Interoperability-](https://www.iab.org/wp-content/IAB-uploads/2016/03/OCF-Derived-Models-for-Interoperability-Between-IoT-Ecosystems_v2-examples.pdf)  
268 [Between-IoT-Ecosystems\\_v2-examples.pdf](https://www.iab.org/wp-content/IAB-uploads/2016/03/OCF-Derived-Models-for-Interoperability-Between-IoT-Ecosystems_v2-examples.pdf)

269 Zigbee Cluster Library Specification, Version 1.0  
270 <http://www.zigbee.org/zigbee-for-developers/zigbee-3-0/>

271 ZigBee Lighting & Occupancy Device, Version 1.0  
272 <http://www.zigbee.org/zigbee-for-developers/zigbee-3-0/>

273 **3 Terms, definitions symbols and abbreviations**

274 For the purposes of this document, the terms and definitions given in ISO/IEC 30118-1:2018,  
275 ISO/IEC 30118-2:2019, and ISO/IEC 30118-3:2019 and the following apply.

276 ISO and IEC maintain terminological databases for use in standardization at the following  
277 addresses:

278 – ISO Online browsing platform: available at <https://www.iso.org/obp>

279 – IEC Electropedia: available at <http://www.electropedia.org/>

## 280 **4 Document conventions and organization**

### 281 **4.1 Conventions**

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

### 286 **4.2 Notation**

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

289 Required (or shall or mandatory).

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

293 Recommended (or should).

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

301 Allowed (or allowed).

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

305 Conditionally allowed (CA)

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

308 Conditionally required (CR)

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

312 DEPRECATED

313 Although these features are still described in this document, they should not be implemented  
314 except for backward compatibility. The occurrence of a deprecated feature during operation of  
315 an implementation compliant with the current document has no effect on the implementation’s  
316 operation and does not produce any error conditions. Backward compatibility may require that  
317 a feature is implemented and functions as specified but it shall never be used by  
318 implementations compliant with this document.

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

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

## 321 **5 Theory of Operation**

### 322 **5.1 Interworking Approach**

323 The interworking between ZigBee Clusters and OCF defined Resources is modelled using the  
324 derived model syntax described in Derived Models for Interoperability between IoT Ecosystems.

### 325 **5.2 Mapping Syntax**

#### 326 **5.2.1 Introduction**

327 Within the defined syntax for derived modelling used by this document there are two blocks that  
328 define the actual Property-Property equivalence or mapping. These blocks are identified by the  
329 keywords "x-to-ocf" and "x-from-ocf". Derived Models for Interoperability between IoT Ecosystems  
330 does not define a rigid syntax for these blocks; they are free form string arrays that contain pseudo-  
331 coded mapping logic.

332 Within this document we apply the rules in defined in clause 5.2 to these blocks to ensure  
333 consistency and re-usability and extensibility of the mapping logic that is defined.

#### 334 **5.2.2 General**

335 All statements are terminated with a carriage return.

#### 336 **5.2.3 Value Assignment**

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

#### 339 **5.2.4 Property Naming**

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

343 The name of the OCF defined Property is prepended by the ecosystem designator "ocf" to avoid  
344 ambiguity (e.g. "ocf.step")

#### 345 **5.2.5 Range**

346 The range on the OCF side is fixed.

#### 347 **5.2.6 Arrays**

348 An array element is indicated by the use of square brackets "[]" with the index of the element  
349 contained therein, e.g. range [1]. All arrays start at an index of 0.

#### 350 **5.2.7 Default Mapping**

351 There are cases where the specified mapping is not possible as one or more of the Properties  
352 being mapped is optional in the source model. In all such instances a default mapping is provided.  
353 (e.g. "transitiontime = 1")

#### 354 **5.2.8 Conditional Mapping**

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

356 If "condition", then "mapping".

357 is applied.

358 E.g. if onoff = false, then ocf.value = false

359 **5.2.9 Method Invocation**

360 The invocation of a command from the derived ecosystem as part of the mapping from an OCF  
361 Resource is indicated by the use of a double colon ":" delimiter between the applicable resource,  
362 service, interface or other construct identifier and the command name. The command name always  
363 includes trailing parentheses which would include any parameters should they be passed.

364 For example when dealing with the "on()" command for Zigbee On/off Cluster this gives a complete  
365 command invocation as: "zb.command.onoff::on()".

366 **6 Device Type Mapping**

367 **6.1 Introduction**

368 This clause contains the mappings from Zigbee Device Types to OCF Device Types.

369 **6.2 Zigbee Device Types to OCF Device Types**

370 Table 1 captures the equivalency mapping between Zigbee defined Device Types (Please see  
371 reference Zigbee Cluster Library Specification) and OCF defined Device Types (please see  
372 reference ISO/IEC 30118-5:2019).

373 **Table 1 – Zigbee to OCF Device Type Mapping**

<b>Zigbee Device Type</b>	<b>Zigbee Device ID</b>	<b>OCF Device Type</b>
<b>On/off Output</b>	0x0002	oic.d.smartplug
<b>Mains Power Outlet</b>	0x0009	oic.d.smartplug
<b>Smart Plug</b>	0x0051	oic.d.smartplug
<b>On/Off Light</b>	0x0100	oic.d.light
<b>Dimmable Light</b>	0x0101	oic.d.light
<b>Color Dimmable Light</b>	0x0102	oic.d.light
<b>Color Temperature Light</b>	0x010c	oic.d.light
<b>Extended Color Light</b>	0x010d	oic.d.light
<b>Window Covering Device</b>	0x0202	oic.d.blind

<b>Thermostat</b>	0x0301	oic.d.thermostat
<b>Temperature Sensor</b>	0x0302	oic.d.sensor
<b>Occupancy Sensor</b>	0x0107	oic.d.sensor
<b>IAS Zone</b>	0x0402	oic.d.sensor

374 **7 Resource to ZigBee Cluster Equivalence**

375 **7.1 Introduction**

376 This clause introduces new Resource Types for mapping between Zigbee Clusters and OCF  
 377 Resources and lists the complete set of applicable Zigbee Clusters and equivalent OCF Resource  
 378 Type(s) in clause 7.2

379 **7.2 Zigbee Clusters to OCF Resources**

380 **7.2.1 Introduction**

381 Table 2 captures the equivalency mapping between Zigbee defined Clusters (see Zigbee Cluster  
 382 Library Specification) and OCF defined Resource Types (see ISO/IEC 30118-4:2019). Detailed  
 383 Property by Property mappings are provided in clause 7.1.

384 clause 8 captures the mappings for mandatory server clusters for Zigbee 3.0 devices

385 **Table 2 – Zigbee Server Cluster to OCF Resource Type Mapping**

<b>Zigbee Cluster</b>	<b>OCF Resource Type Name</b>	<b>OCF Resource Type ID</b>	<b>OCF Interface(s)</b>
<b>On/off</b>	Binary Switch	oic.r.switch.binary	oic.if.a
<b>Level Control</b>	Dimming	oic.r.light.dimming	oic.if.a
<b>Color Control</b>	Colour Hue and Saturation, Colour Space Coordinates, Colour Temperature	oic.r.colour.hs, oic.r.colour.csc, oic.r.colour.colourtemperature,	oic.if.a
<b>Thermostat</b>	Temperature (3)	oic.r.temperature (3) * 1 for sensor, 2 for heater and cooler	oic.if.s oic.if.a
<b>Window Covering</b>	Window Covering	oic.r.windowcovering, oic.r.openlevel (4)	oic.if.rw oic.if.a

		* 2 for lift (percentage scale and c m scale), 2 for tilt (percentage scale and c m scale)	
<b>Temperature Measurement</b>	Temperature	oic.r.temperature	oic.if.s
<b>Occupancy Sensing</b>	Presence Sensor	oic.r.sensor.presence	oic.if.s
<b>IAS Zone</b>	IAS Zone	oic.r.ias.zone	oic.if.rw

386

### 387 **7.2.2 On/off**

388 The APIs with "zcl.onoff" define the mapping between an instance of an OCF Binary Switch  
389 Resource and the Zigbee On/off Cluster. In clause 8.14 a RETRIEVE on an OCF Binary Switch  
390 Resource maps to a general Read command on a Zigbee On/off Cluster. The value of Zigbee  
391 Attribute in Zigbee On/off Cluster is retrieved via the general Read command and mapped with the  
392 value of OCF Property in OCF Binary Switch Resource. In clause 8.13 an UPDATE on a Binary  
393 Switch maps to a command invocation on either "on()" command or "off()" command of Zigbee  
394 On/off Cluster. "value = true" maps to "on()", "value = false" maps to "off()" of Zigbee On/off Cluster.

### 395 **7.2.3 Level Control**

396 The APIs with "zcl.levelcontrol" define the mapping between an instance of an OCF Dimming  
397 Resource and the Zigbee Level Control Cluster. In clause 8.11, a RETRIEVE on an OCF Dimming  
398 Resource maps to a general Read command on a Zigbee Level Control Cluster. The value of  
399 Zigbee Attribute in Zigbee Level Control Cluster is retrieved via the general Read command and  
400 mapped with the value of OCF Property in OCF Dimming Resource. In clause 8.10, an UPDATE  
401 on a "dimmingSetting" maps to a command invocation on "movetolevel(level,transitiontime=0)" of  
402 Zigbee Level Control Cluster.

### 403 **7.2.4 Color Control**

404 The APIs with "zcl.colorcontrol" define the mapping between instances of OCF Colour Resources  
405 and the Zigbee Color Control Cluster. The OCF Colour Resources are OCF Hue and Saturation  
406 Resource, OCF Colour Space Coordinate Resource, OCF Colour Temperature Resource.

407 The APIs with "zcl.colorcontrol\_hs" define the mapping between an instance of OCF Hue and  
408 Saturation Resources and the Zigbee Color Control Cluster. In clause 8.7, a RETRIEVE on an are  
409 OCF Hue and Saturation Resource maps to a general Read command on a Zigbee Color Control  
410 Cluster. The values of Zigbee Attributes in Zigbee Color Control Cluster are retrieved via the  
411 general Read command and mapped with those of OCF Properties in OCF Hue and Saturation  
412 Resource. In clause 8.6, an UPDATE on OCF Colour Hue and Saturation Resource maps to a  
413 command invocation on "movetohueandsaturation(hue,saturation,transitiontime=0)" of Zigbee  
414 Color Control Cluster.

415 The APIs with "zcl.colorcontrol\_csc" define the mapping between an instance of OCF Colour Space  
416 Coordinate Resource and the Zigbee Color Control Cluster. In clause 8.3, a RETRIEVE on an OCF  
417 Colour Space Coordinate Resource maps to a general Read command on a Zigbee Color Control  
418 Cluster. The values of Zigbee Attributes in Zigbee Color Control Cluster are retrieved via the  
419 general Read command and mapped with those of OCF Properties in OCF Colour Space  
420 Coordinate Resource. In clause 8.2, an UPDATE on OCF Colour Space Coordinate Resource

421 maps to a command invocation on "movetocolor(colorx,colory,transitiontime=0)" of Zigbee Color  
422 Control Cluster.

423 The APIs with "zcl.colorcontrol\_ct" define the mapping between an instance of OCF Colour  
424 Temperature Resource and the Zigbee Color Control Cluster. In clause 8.4, a RETRIEVE on an  
425 OCF Colour Temperature Resource maps to a general Read command on a Zigbee Color Control  
426 Cluster. The values of Zigbee Attributes in Zigbee Color Control Cluster are retrieved via the  
427 general Read command and mapped with those of OCF Properties in OCF Colour Temperature  
428 Resource. In clause 8.5, an UPDATE on OCF Colour Temperature Resource maps to a command  
429 invocation on "movetocolor(temperature,transitiontime=0)" of Zigbee Color  
430 Control Cluster.

### 431 **7.2.5 Thermostat**

432 The APIs with "zcl.thermostat" define the mapping between 3 instances of OCF Temperature  
433 Resources and the Zigbee Thermostat Cluster. The 3 instances of OCF Temperature Resources  
434 are for sensor, heater, and cooler respectively.

435 The API with "zcl.thermostat\_currenttemperature" defines the mapping between an instance of  
436 OCF Temperature Resource and the Zigbee Thermostat Cluster for sensor. In clause 8.17, a  
437 RETRIEVE on an OCF Temperature Resource maps to a general Read command on a Zigbee  
438 Thermostat Cluster. The value of Zigbee Attribute in Zigbee Thermostat Cluster is retrieved via the  
439 general Read command and mapped with the value of OCF Property in OCF Temperature  
440 Resource. The value represents the current temperature.

441 The API with "zcl.thermostat\_heat" defines the mapping between an instance of OCF Temperature  
442 Resource and the Zigbee Thermostat Cluster for heater. In clause 8.18, an UPDATE on  
443 "temperature" of OCF Temperature Resource maps to "setpointraiselower(mode=heat mode,  
444 amount)" on a Zigbee Thermostat Cluster.

445 The API with "zcl.thermostat\_cool" defines the mapping between an instance of OCF Temperature  
446 Resource and the Zigbee Thermostat Cluster for cooler. In clause 8.16, an UPDATE on  
447 "temperature" of OCF Temperature Resource maps to "setpointraiselower(mode=cool mode,  
448 amount)" on a Zigbee Thermostat Cluster.

### 449 **7.2.6 Window Covering**

450 The APIs with "zcl.windowcovering" define the mapping between 5 instances of OCF Resources  
451 and the Zigbee Window Covering Cluster. The 5 instances of OCF Resources are the instance of  
452 OCF Window Covering Resource and the 4 instances of OCF Open Level Resources. The 4  
453 instances of OCF Open Level Resources are for lift level with percentage scale, lift level with  
454 centimetre scale, tilt level with percentage scale, tilt level with centimetre scale.

455 The API with "zcl.windowcovering\_conf" defines the mapping between an instance of OCF Window  
456 Covering Resource and the Zigbee window Covering Cluster. In clause 8.20, a RETRIEVE on an  
457 OCF Window Covering Resource maps to a general Read command on a Zigbee Window Covering  
458 Cluster. The values of Zigbee Attributes in Zigbee Window Covering Cluster is retrieved via the  
459 general Read command and mapped with the value of OCF Property in OCF Window Covering  
460 Resource. In clause 8.19, an UPDATE on OCF Window Covering Resource maps to a general  
461 Write command on a Zigbee Window Covering Cluster.

462 The API with "zcl.windowcovering\_liftpercentage" defines the mapping between an instance of  
463 OCF Open Level Resource and the Zigbee window Covering Cluster for lift with percentage scale.  
464 In clause 8.22, a RETRIEVE on an OCF Open Level Resource maps to a general Read command  
465 on "CurrentPositionLiftPercentage" of Zigbee Window Covering Cluster. The value of Zigbee  
466 Attribute in Zigbee Window Covering Cluster is retrieved via the general Read command and  
467 mapped with the value of OCF Property in OCF Open Level Resource. In clause 8.21, an UPDATE



468 on OCF Open Level Resource maps to "gotoliftpercentage(percentageliftvalue)" on a Zigbee  
469 Window Covering Cluster.

470 The API with "zcl.windowcovering\_liftposition" defines the mapping between an instance of OCF  
471 Open Level Resource and the Zigbee window Covering Cluster for lift with centimetre scale. In  
472 clause 8.24, a RETRIEVE on an OCF Open Level Resource maps to a general Read command on  
473 "CurrentPosition-Lift" of Zigbee Window Covering Cluster. The value of Zigbee Attribute in Zigbee  
474 Window Covering Cluster is retrieved via the general Read command and mapped with the value  
475 of OCF Property in OCF Open Level Resource. In clause 8.23, an UPDATE on OCF Open Level  
476 Resource maps to "gotoliftvalue(liftvalue)" on a Zigbee Window Covering Cluster.

477 The API with "zcl.windowcovering\_tiltpercentage" defines the mapping between an instance of  
478 OCF Open Level Resource and the Zigbee window Covering Cluster for tilt with percentage scale.  
479 In clause 8.26, a RETRIEVE on an OCF Open Level Resource maps to a general Read command  
480 on "CurrentPositionTiltPercentage" of Zigbee Window Covering Cluster. The value of Zigbee  
481 Attribute in Zigbee Window Covering Cluster is retrieved via the general Read command and  
482 mapped with the value of OCF Property in OCF Open Level Resource. In clause 8.25, an UPDATE  
483 on OCF Open Level Resource maps to "gototiltpercentage(percentagetiltvalue)" on a Zigbee  
484 Window Covering Cluster.

485 The API with "zcl.windowcovering\_tiltposition" defines the mapping between an instance of OCF  
486 Open Level Resource and the Zigbee window Covering Cluster for tilt with centimetre scale. In  
487 clause 8.28, a RETRIEVE on an OCF Open Level Resource maps to a general Read command on  
488 "CurrentPosition-Tilt" of Zigbee Window Covering Cluster. The value of Zigbee Attribute in Zigbee  
489 Window Covering Cluster is retrieved via the general Read command and mapped with the value  
490 of OCF Property in OCF Open Level Resource. In clause 8.27, an UPDATE on OCF Open Level  
491 Resource maps to "gototiltvalue(tiltvalue)" on a Zigbee Window Covering Cluster.

#### 492 **7.2.7 Temperature Measurement**

493 The API with "zcl.temperaturemeasurement" defines the mapping between an instance of an OCF  
494 Temperature Resource and the Zigbee Temperature Measurement Cluster for sensor. In clause  
495 8.15, a RETRIEVE on an OCF Temperature Resource maps to a general Read command on a  
496 Zigbee Temperature Measurement Cluster. The value of Zigbee Attribute in Zigbee Temperature  
497 Measurement Cluster is retrieved via the general Read command and mapped with the value of  
498 OCF Property in OCF Temperature Resource. The value represents the current temperature.

#### 499 **7.2.8 Occupancy Sensing**

500 The API with "zcl.occupancysensing" defines the mapping between an instance of an OCF  
501 Presence Sensor Resource and the Zigbee Occupancy Sensing Cluster. In clause 8.12, a  
502 RETRIEVE on an OCF Presence Sensor Resource maps to a general Read command on a Zigbee  
503 Occupancy Sensing Cluster. The value of Zigbee Attribute in Zigbee Occupancy Sensing Cluster  
504 is retrieved via the general Read command and mapped with the value of OCF Property in OCF  
505 Presence Sensor.

#### 506 **7.2.9 IAS Zone**

507 The API with "zcl.iaszone" defines the mapping between an instance of an OCF IAS Zone  
508 Resource and the Zigbee IAS Zone Cluster. In clause 8.9, a RETRIEVE on an IAS Zone Resource  
509 maps to a general Read command on a Zigbee IAS Zone Cluster. The values of Zigbee Attributes  
510 in Zigbee IAS Zone Cluster are retrieved via the general Read command and mapped with those  
511 of OCF Properties in OCF IAS Zone Resource. In clause 8.8, an UPDATE on OCF IAS Zone  
512 Resource maps to a general Write command on a Zigbee IAS Zone Cluster.

513 **8 Detailed Mapping APIs**

514 **8.1 below**

515 **8.2 Introduction**

516 This clause provides an API and mapping description that aligns with the Derived Modelling syntax  
 517 described in Derived Models for Interoperability between IoT Ecosystems for all Module Classes  
 518 and Resources that are within scope.

519 The derived model definitions presented in clause 8 are formatted for readability, and so may  
 520 appear to have extra line breaks.

521 **8.3 Color Control Cluster - Color Space - Control**

522 **8.3.1 Derived model**

523 The derived model: "zcl.colorcontrol\_csc.control.movetocolor".

524 **8.3.2 Property definition**

525 Table 3 provides the detailed per Property mapping for "zcl.colorcontrol\_csc.control.movetocolor".

526 **Table 3 – The Property mapping for "zcl.colorcontrol\_csc.control.movetocolor".**

Zigbee Property name	OCF Resource	To OCF	From OCF
colory	oic.r.colour.csc	N/A	colory= ocf.csc[1]*65536 & transitiontime=0zcl.command.colorcontrol::movetocolor(colorx,colory,transitiontime).
colorx	oic.r.colour.csc	N/A	colorx =ocf.csc[0]*65536 & transitiontime=0zcl.command.colorcontrol::movetocolor(colorx,colory,transitiontime).

527 Table 4 provides the details of the Properties that are part of  
 528 "zcl.colorcontrol\_csc.control.movetocolor".

529 **Table 4 – The Properties of "zcl.colorcontrol\_csc.control.movetocolor".**

Zigbee name	Property	Type	Required	Description
colory		number	no	Move to certain value(s) of color coordinates as fast as possible with transitiontime=0. transitiontime is set by Zigbee 3.0 translator.
colorx		number	no	Move to certain value(s) of color coordinates as fast as possible with transitiontime=0. transitiontime is set by Zigbee 3.0 translator.

530 **8.3.3 Derived model definition**

```

531 {
532   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_csc.control.json#",
533   "$schema": "http://json-schema.org/draft-04/schema#",
534   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
535   "title": "Color Control Cluster - Color Space - Control",
536   "definitions": {
537     "zcl.colorcontrol_csc.control.movetocolor": {
538       "properties": {
539         "colorx": {
540           "type": "number",
541           "description": "Move to certain value(s) of color coordinates as fast as possible with
542 transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
543           "x-ocf-conversion": {
544             "x-ocf-alias": "oic.r.colour.csc",
545             "x-from-ocf": [
546               "colorx =ocf.csc[0]*65536 & transitiontime=0",
547               "zcl.command.colorcontrol::movetocolor(colorx,colory,transitiontime)."
548             ],
549             "x-to-ocf": [
550               "N/A"
551             ]
552           }
553         },
554         "colory": {
555           "type": "number",
556           "description": "Move to certain value(s) of color coordinates as fast as possible with
557 transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
558           "x-ocf-conversion": {
559             "x-ocf-alias": "oic.r.colour.csc",
560             "x-from-ocf": [
561               "colory= ocf.csc[1]*65536 & transitiontime=0",
562               "zcl.command.colorcontrol::movetocolor(colorx,colory,transitiontime)."
563             ],
564             "x-to-ocf": [
565               "N/A"
566             ]
567           }
568         }
569       }
570     }
571   },
572   "type": "object",
573   "allOf": [
574     {"$ref": "#/definitions/zcl.colorcontrol_csc.control.movetocolor"}
575   ]
576 }
577

```

578 **8.4 Color Control Cluster - Color Space - Information**

579 **8.4.1 Derived model**

580 The derived model: "zcl.colorcontrol\_csc.info".

581 **8.4.2 Property definition**

582 Table 5 provides the detailed per Property mapping for "zcl.colorcontrol\_csc.info".

583 **Table 5 – The Property mapping for "zcl.colorcontrol\_csc.info".**

Zigbee name	Property	OCF Resource	To OCF	From OCF
currentX		oic.r.colour.csc	ocf.csc[0] currentX/65536	= N/A
currentY		oic.r.colour.csc	ocf.csc[1] currentY/65536	= N/A

584 Table 6 provides the details of the Properties that are part of "zcl.colorcontrol\_csc.info".

585 **Table 6 – The Properties of "zcl.colorcontrol\_csc.info".**

Zigbee name	Property	Type	Required	Description
currentX		integer	no	current value of the normalized chromaticity value x, as defined in the CIE xy Color Space
currentY		integer	no	current value of the normalized chromaticity value y, as defined in the CIE xy Color Space

586 **8.4.3 Derived model definition**

```

587 {
588   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_csc.info.json#",
589   "$schema": "http://json-schema.org/draft-04/schema#",
590   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
591   "title": "Color Control Cluster - Color Space - Information",
592   "definitions": {
593     "zcl.colorcontrol_csc.info": {
594       "type": "object",
595       "properties": {
596         "currentX": {
597           "type": "integer",
598           "description": "current value of the normalized chromaticity value x, as defined in the
599 CIE xy Color Space",
600           "x-ocf-conversion": {
601             "x-ocf-alias": "oic.r.colour.csc",
602             "x-to-ocf": [
603               "ocf.csc[0] = currentX/65536"
604             ],
605             "x-from-ocf": [
606               "N/A"
607             ]
608           }
609         },
610         "currentY": {
611           "type": "integer",
612           "description": "current value of the normalized chromaticity value y, as defined in the
613 CIE xy Color Space",
614           "x-ocf-conversion": {
615             "x-ocf-alias": "oic.r.colour.csc",
616             "x-to-ocf": [
617               "ocf.csc[1] = currentY/65536"
618             ],
619             "x-from-ocf": [
620               "N/A"
621             ]
622           }
623         }
624       }
625     }
626   },
627   "type": "object",
628   "allOf": [
629     {"$ref": "#/definitions/zcl.colorcontrol_csc.info"}
630   ],
631   "required": ["currentx", "currenty"]
632 }
633

```

634 **8.5 Color Control Cluster - Color Temperature - Information**

635 **8.5.1 Derived model**

636 The derived model: "zcl.colorcontrol\_ct.control.movetocolortemperature".

637 **8.5.2 Property definition**

638 Table 7 provides the detailed per Property mapping for  
639 "zcl.colorcontrol\_ct.control.movetocolortemperature".

640 **Table 7 – The Property mapping for "zcl.colorcontrol\_ct.control.movetocolortemperature".**

Zigbee Property name	OCF Resource	T o O C F	From OCF
colortemperature	oic.r.colour.colourtemperature	N/A	colourtemperature=ocf.ct & transitiontime=0zcl.command.colorcontrol::movetocolortemperature(colortemperature,transitiontime)

641 Table 8 provides the details of the Properties that are part of  
642 "zcl.colorcontrol\_ct.control.movetocolortemperature".

643 **Table 8 – The Properties of "zcl.colorcontrol\_ct.control.movetocolortemperature".**

Zigbee Property name	Type	Required	Description
colortemperature	integer	no	Move to certain value of colortemperature as fast as possible with transitiontime=0. transitiontime is set by Zigbee 3.0 translator.

644 **8.5.3 Derived model definition**

```

645 {
646   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_ct.control.json#",
647   "$schema": "http://json-schema.org/draft-04/schema#",
648   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
649   "title": "Color Control Cluster - Color Temperature - Information",
650   "definitions": {
651     "zcl.colorcontrol_ct.control.movetocolortemperature": {
652       "properties": {
653         "colortemperature": {
654           "type": "integer",
655           "description": "Move to certain value of colortemperature as fast as possible with
656 transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
657           "x-ocf-conversion": {
658             "x-ocf-alias": "oic.r.colour.colourtemperature",
659             "x-from-ocf": [
660               "colourtemperature=ocf.ct & transitiontime=0",
661               "zcl.command.colorcontrol::movetocolortemperature(colortemperature,transitiontime)"
662             ],
663             "x-to-ocf": [
664               "N/A"
665             ]
666           }
667         }
668       }
669     },
670     "type": "object",
671     "allOf": [
672

```

```

673     {"$ref": "#/definitions/zcl.colorcontrol_ct.control.movetocolortemperature"}
674   ]
675 }
676

```

## 677 8.6 Color Control Cluster - Color Temperature - Information

### 678 8.6.1 Derived model

679 The derived model: "zcl.colorcontrol\_ct.info".

### 680 8.6.2 Property definition

681 Table 9 provides the detailed per Property mapping for "zcl.colorcontrol\_ct.info".

682 **Table 9 – The Property mapping for "zcl.colorcontrol\_ct.info".**

Zigbee name	Property	OCF Resource	To OCF	From OCF
colorphysicalmax		oic.r.colour.colourtemperature	ocf.range[1] colorphysicalmax	= N/A
colortemphysicalmin		oic.r.colour.colourtemperature	ocf.range[0] colortemphysicalmin	= N/A
colortemperaturemired		oic.r.colour.colourtemperature	ocf.ct colortemperaturemired	= N/A

683 Table 10 provides the details of the Properties that are part of "zcl.colorcontrol\_ct.info".

684 **Table 10 – The Properties of "zcl.colorcontrol\_ct.info".**

Zigbee name	Property	Type	Required	Description
colorphysicalmax		integer	no	maximum mired value supported by the hardware
colortemphysicalmin		integer	no	minimum mired value supported by the hardware
colortemperaturemired		integer	yes	Scaled inverse of the current value of the color temperature

### 685 8.6.3 Derived model definition

```

686 {
687   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_ct.info.json#",
688   "$schema": "http://json-schema.org/draft-04/schema#",
689   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
690   "title": "Color Control Cluster - Color Temperature - Information",
691   "definitions": {
692     "zcl.colorcontrol_ct.info": {
693       "type": "object",
694       "properties": {
695         "colortemperaturemired": {
696           "type": "integer",
697           "description": "Scaled inverse of the current value of the color temperature",
698           "x-ocf-conversion": {
699             "x-ocf-alias": "oic.r.colour.colourtemperature",
700             "x-to-ocf": [
701               "ocf.ct = colortemperaturemired"
702             ],
703             "x-from-ocf": [
704               "N/A"
705             ]
706           }
707         }
708       }
709     }
710   }

```

```

708     "colortempphysicalmin": {
709       "type": "integer",
710       "description": "minimum mired value supported by the hardware",
711       "x-ocf-conversion": {
712         "x-ocf-alias": "oic.r.colour.colourtemperature",
713         "x-to-ocf": [
714           "ocf.range[0] = colortempphysicalmin"
715         ],
716         "x-from-ocf": [
717           "N/A"
718         ]
719       }
720     },
721     "colorphysicalmax": {
722       "type": "integer",
723       "description": "maximum mired value supported by the hardware",
724       "x-ocf-conversion": {
725         "x-ocf-alias": "oic.r.colour.colourtemperature",
726         "x-to-ocf": [
727           "ocf.range[1] = colorphysicalmax"
728         ],
729         "x-from-ocf": [
730           "N/A"
731         ]
732       }
733     }
734   }
735 }
736 },
737 "type": "object",
738 "allof": [
739   {"$ref": "#/definitions/zcl.colorcontrol_ct.info"}
740 ],
741 "required": ["colortemperaturemired", "colortempphysicalmin", "colortempphysicalmax" ]
742 }
743

```

## 744 8.7 Color Control Cluster - Hue and Saturation - Control

### 745 8.7.1 Derived model

746 The derived model: "zcl.colorcontrol\_hs.control.movetohueandsaturation".

### 747 8.7.2 Property definition

748 Table 11 provides the detailed per Property mapping for  
749 "zcl.colorcontrol\_hs.control.movetohueandsaturation".

750 **Table 11 – The Property mapping for**  
751 **"zcl.colorcontrol\_hs.control.movetohueandsaturation".**

Zigbee Property name	OCF Resource	To OCF	From OCF
saturation	oic.r.colour.hs	N/A	saturation=ocf.saturation & transitiontime=0zcl.command.colorcontrol::movetohueandsaturation (hue,saturation,transitiontime)
hue	oic.r.colour.hs	N/A	hue=ocf.hue/360 * 254 & transitiontime=0zcl.command.colorcontrol::movetohueandsaturation (hue,saturation,transitiontime)

752 Table 12 provides the details of the Properties that are part of  
753 "zcl.colorcontrol\_hs.control.movetohueandsaturation".

Table 12 – The Properties of "zcl.colorcontrol\_hs.control.movetohueandsaturation".

Zigbee name	Property	Type	Required	Description
saturation		integer	no	Move to certain value(s) of hue or saturation or both as fast as possible with transitiontime=0. transitiontime is set by Zigbee 3.0 translator.
hue		integer	no	Move to certain value(s) of hue or saturation or both as fast as possible with transitiontime=0. transitiontime is set by Zigbee 3.0 translator.

### 755 8.7.3 Derived model definition

```

756 {
757   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_hs.control.json#",
758   "$schema": "http://json-schema.org/draft-04/schema#",
759   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
760   "title": "Color Control Cluster - Hue and Saturation - Control",
761   "definitions": {
762     "zcl.colorcontrol_hs.control.movetohueandsaturation": {
763       "properties": {
764         "hue": {
765           "type": "integer",
766           "description": "Move to certain value(s) of hue or saturation or both as fast as possible
767 with transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
768           "x-ocf-conversion": {
769             "x-ocf-alias": "oic.r.colour.hs",
770             "x-from-ocf": [
771               "hue=ocf.hue/360 * 254 & transitiontime=0",
772               "zcl.command.colorcontrol::movetohueandsaturation(hue,saturation,transitiontime)"
773             ],
774             "x-to-ocf": [
775               "N/A"
776             ]
777           }
778         },
779         "saturation": {
780           "type": "integer",
781           "description": "Move to certain value(s) of hue or saturation or both as fast as possible
782 with transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
783           "x-ocf-conversion": {
784             "x-ocf-alias": "oic.r.colour.hs",
785             "x-from-ocf": [
786               "saturation=ocf.saturation & transitiontime=0",
787               "zcl.command.colorcontrol::movetohueandsaturation(hue,saturation,transitiontime)"
788             ],
789             "x-to-ocf": [
790               "N/A"
791             ]
792           }
793         }
794       }
795     }
796   },
797   "type": "object",
798   "allOf": [
799     {"$ref": "#/definitions/zcl.colorcontrol_hs.control.movetohueandsaturation"}
800 ]

```



801 }  
802

## 803 8.8 Color Control Cluster - Hue and Saturation - Information

### 804 8.8.1 Derived model

805 The derived model: "zcl.colorcontrol\_hs.info".

### 806 8.8.2 Property definition

807 Table 13 provides the detailed per Property mapping for "zcl.colorcontrol\_hs.info".

808 **Table 13 – The Property mapping for "zcl.colorcontrol\_hs.info".**

Zigbee name	Property	OCF Resource	To OCF	From OCF
currentsaturation		oic.r.colour.hs	ocf.saturation = currentsaturation & maximumsaturation=254	N/A
currenthue		oic.r.colour.hs	ocf.hue = currenthue/254 * 360	N/A

809 Table 14 provides the details of the Properties that are part of "zcl.colorcontrol\_hs.info".

810 **Table 14 – The Properties of "zcl.colorcontrol\_hs.info".**

Zigbee name	Property	Type	Required	Description
currentsaturation		integer	yes	current saturation value of the light
currenthue		integer	yes	current hue value of the light

### 811 8.8.3 Derived model definition

```
812 {  
813   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_hs.info.json#",  
814   "$schema": "http://json-schema.org/draft-04/schema#",  
815   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",  
816   "title": "Color Control Cluster - Hue and Saturation - Information",  
817   "definitions": {  
818     "zcl.colorcontrol_hs.info": {  
819       "type": "object",  
820       "properties": {  
821         "currenthue": {  
822           "type": "integer",  
823           "description": "current hue value of the light",  
824           "x-ocf-conversion": {  
825             "x-ocf-alias": "oic.r.colour.hs",  
826             "x-to-ocf": [  
827               "ocf.hue = currenthue/254 * 360"  
828             ]  
829           },  
830           "x-from-ocf": [  
831             "N/A"  
832           ]  
833         },  
834         "currentsaturation": {  
835           "type": "integer",  
836           "description": "current saturation value of the light",  
837           "x-ocf-conversion": {  
838             "x-ocf-alias": "oic.r.colour.hs",  
839             "x-to-ocf": [  
840               "ocf.saturation = currentsaturation & maximumsaturation=254"  
841             ]  
842           },  
843           "x-from-ocf": [  
844             "N/A"  
845           ]  
846         }  
847       }  
848     }  
849   }  
850 }
```

```

843         "N/A"
844     ]
845 }
846 }
847 }
848 }
849 },
850 "type": "object",
851 "allOf": [
852   {"$ref": "#/definitions/zcl.colorcontrol_hs.info"}
853 ],
854 "required": ["currenthue", "currentsaturation"]
855 }
856

```

## 8.9 IAS Zone Cluster - Control

### 8.9.1 Derived model

The derived model: "zcl.iaszone.control".

### 8.9.2 Property definition

Table 15 provides the detailed per Property mapping for "zcl.iaszone.control".

**Table 15 – The Property mapping for "zcl.iaszone.control".**

Zigbee Property name	OCF Resource	To OCF	From OCF
currentzonesensitivitylevel	oic.r.ias.zone	N/A	currentzonesensitivitylevel = ocf.currentzonesensitivitylevelzcl.command.general::write(currentzonesensitivitylevel)

Table 16 provides the details of the Properties that are part of "zcl.iaszone.control".

**Table 16 – The Properties of "zcl.iaszone.control".**

Zigbee Property name	Type	Required	Description
currentzonesensitivitylevel	integer	no	Set a sensitivity level of IAS Zone

### 8.9.3 Derived model definition

```

866 {
867   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.iaszone.control.json#",
868   "$schema": "http://json-schema.org/draft-04/schema#",
869   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
870   "title": "IAS Zone Cluster - Control",
871   "definitions": {
872     "zcl.iaszone.control": {
873       "properties": {
874         "currentzonesensitivitylevel": {
875           "type": "integer",
876           "description": "Set a sensitivity level of IAS Zone",
877           "x-ocf-conversion": {
878             "x-ocf-alias": "oic.r.ias.zone",
879             "x-from-ocf": [
880               "currentzonesensitivitylevel = ocf.currentzonesensitivitylevel",
881               "zcl.command.general::write(currentzonesensitivitylevel)"
882             ],
883             "x-to-ocf": [
884               "N/A"
885             ]
886           }
887         }
888       }
889     }
890   }

```

```

890     },
891     "type": "object",
892     "allOf": [
893         {"$ref": "#/definitions/zcl.iaszone.control"}
894     ]
895 }
896

```

## 8.10 IAS Zone Cluster - Information

### 8.10.1 Derived model

The derived model: "zcl.iaszone.info".

### 8.10.2 Property definition

Table 17 provides the detailed per Property mapping for "zcl.iaszone.info".

**Table 17 – The Property mapping for "zcl.iaszone.info".**

Zigbee Property name	OCF Resource	To OCF	From OCF
zoneID	oic.r.iaszone	ocf.zoneid=zoneID	N/A
numberofzonesensitivitylevels_supported	oic.r.iaszone	ocf.numzonesensitivitylevel=numberofzonesensitivitylevels_supported	N/A
zonestate	oic.r.iaszone	if zonestate=0x00, ocf.zonestate=false if zonestate=0x01, ocf.zonestate=true	N/A
IAS_CIE_address	oic.r.iaszone	ocf.iascieaddress= IAS_CIE_address	N/A
zonetype	oic.r.iaszone	if zonetype=0x0000, ocf.zonetype=Standard CIE if zonetype=0x000d, ocf.zonetype=Motion sensor if zonetype=0x0015, ocf.zonetype=Contact switch if zonetype=0x0028, ocf.zonetype=Fire sensor if zonetype=0x002a, ocf.zonetype=Water sensor if zonetype=0x002b, ocf.zonetype=Carbon Monoxide (CO) sensor if zonetype=0x002c, ocf.zonetype=Personal emergency device if zonetype=0x002d, ocf.zonetype=Vibration/Movement sensor if zonetype=0x010f, ocf.zonetype=Remote Control if zonetype=0x0115, ocf.zonetype=Keypad if zonetype=0x021d, ocf.zonetype=Keypad Warning Device if zonetype=0x0225, ocf.zonetype=Standard Warning Device if zonetype=0x0226, ocf.zonetype=Glass break sensor if zonetype=0x0229, ocf.zonetype=Security repeater if zonetype=0xffff, ocf.zonetype=Invalid Zone Type	N/A
zonestatus	oic.r.iaszone	if zonetype=0x0000 & zonestatus=xxxxxxxxxx0, ocf.zonestatus.alarms=[''] if zonetype=0x0000 & zonestatus=xxxxxxxxxx1, ocf.zonestatus.alarms=['system'] if zonetype=0x0000 & zonestatus=xxxxxxxxxx0x, ocf.zonestatus.alarms=[''] if zonetype=0x0000 &	N/A

		<pre> zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=["]if zonetype=0x000d &amp; zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=["]if zonetype=0x000d &amp; zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=['intrusion']if zonetype=0x000d                                &amp; zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0x000d &amp; zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=['presence']if zonetype=0x000d                                &amp; zonestatus=xxxxxxxxxxx11, ocf.zonestatus.alarms=['intrusion','presence']if zonetype=0x0015                                &amp; zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=["]if zonetype=0x0015 &amp; zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=['1stportalopenclose']if zonetype=0x0015                                &amp; zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0x0015 &amp; zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=['2ndportalopenclose']if zonetype=0x0015                                &amp; zonestatus=xxxxxxxxxxx11, ocf.zonestatus.alarms=['1stportalopenclose','2n dportalopenclose']if zonetype=0x0028      &amp; zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=["]if zonetype=0x0028 &amp; zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=['fire']if zonetype=0x0028                                &amp; zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0x0028 &amp; zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=["]if zonetype=0x002a &amp; zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=["]if zonetype=0x002a &amp; zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=['wateroverflow']if zonetype=0x002a                                &amp; zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0x002a &amp; zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=["]if zonetype=0x002b &amp; zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=["]if zonetype=0x002b &amp; zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=['CO']if zonetype=0x002b                                &amp; zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0x002b &amp; zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=['cooking']if zonetype=0x002b                                &amp; </pre>	
--	--	---	--

		zonestatus=xxxxxxxxxxx11, ocf.zonestatus.alarms=['CO','cooking']if zonetype=0x002c & zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=['']if zonetype=0x002c & zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=['fall']if zonetype=0x002c & zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=['']if zonetype=0x002c & zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=['emergencybutton']if zonetype=0x002c & zonestatus=xxxxxxxxxxx11, ocf.zonestatus.alarms=['fall','emergencybutton'] if zonetype=0x002d & zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=['']if zonetype=0x002d & zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=['movement']if zonetype=0x002d & zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=['']if zonetype=0x002d & zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=['vibration']if zonetype=0x002d & zonestatus=xxxxxxxxxxx11, ocf.zonestatus.alarms=['movement','vibration']if zonetype=0x010f & zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=['']if zonetype=0x010f & zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=['panic']if zonetype=0x010f & zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=['']if zonetype=0x010f & zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=['emergency']if zonetype=0x010f & zonestatus=xxxxxxxxxxx11, ocf.zonestatus.alarms=['panic','emergency']if zonetype=0x0115 & zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=['']if zonetype=0x0115 & zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=['panic']if zonetype=0x0115 & zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=['']if zonetype=0x0115 & zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=['emergency']if zonetype=0x0115 & zonestatus=xxxxxxxxxxx11, ocf.zonestatus.alarms=['panic','emergency']if zonetype=0x021d & zonestatus=xxxxxxxxxxx0,	
--	--	---	--

		<pre> ocf.zonestatus.alarms=["]if zonetype=0x021d &amp; zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['panic']if zonetype=0x021d                                &amp; zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0x021d &amp; zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['emergency']if zonetype=0x021d                                &amp; zonestatus=xxxxxxxxxxxx11, ocf.zonestatus.alarms=['panic','emergency']if zonetype=0x0225                                &amp; zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=["]if zonetype=0x0225 &amp; zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['glassbreak']if zonetype=0x0225                                &amp; zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0x0225 &amp; zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=["]if zonetype=0x0226 &amp; zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=["]if zonetype=0x0226 &amp; zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=["]if zonetype=0x0226 &amp; zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0x0226 &amp; zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=["]if zonetype=0x0229 &amp; zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=["]if zonetype=0x0229 &amp; zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=["]if zonetype=0x0229 &amp; zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0x0229 &amp; zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=["]if zonetype=0xffff &amp; zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=["]if zonetype=0xffff &amp; zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=["]if zonetype=0xffff &amp; zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0xffff &amp; zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=["]if zonestatus=xxxxxxxxxxxx0xx, ocf.zonestatus.tamper=falseif zonestatus=xxxxxxxxxxxx1xx, ocf.zonestatus.tamper=trueif zonestatus=xxxxxxxxxxxx0xxx, ocf.zonebattery.charge=100                        &amp; ocf.zonebattery.lowbattery=falseif zonestatus=xxxxxxxxxxxx1xxx, ocf.zonebattery.charge=100                        &amp; ocf.zonebattery.lowbattery=trueif zonestatus=xxxxxxxx00xxxx, </pre>	
--	--	---	--

		ocf.zonestatus.zonestatusreports='none'if zonestatus=xxxxxxx01xxxx, ocf.zonestatus.zonestatusreports='statuschang eonly' if zonestatus=xxxxxxx10xxxx, ocf.zonestatus.zonestatusreports='alarmclearo nly' if zonestatus=xxxxxxx11xxxx, ocf.zonestatus.zonestatusreports='statuschang eandalarmclear'if zonestatus=xxxxxxx0xxxxxx, ocf.zonestatus.fault=falseif zonestatus=xxxxxxx1xxxxxx, ocf.zonestatus.fault=trueif zonestatus=xxxxxx0xxxxxxx, ocf.zonepowersource.powerSources=['AC (Mains) Power'] & ocf.zonepowersource.sourcefault=falseif zonestatus=xxxxxx1xxxxxxx, ocf.zonepowersource.powerSources=['AC (Mains) Power'] & ocf.zonepowersource.sourcefault=trueif zonestatus=xxxxx0xxxxxxx, ocf.zonestatus.test=falseif zonestatus=xxxxx1xxxxxxx, ocf.zonestatus.test=trueif zonestatus=xxxx0xxxxxxx, ocf.zonepowersource.powerSources=['Internal Battery'] & oic.r.ias.zone.zonebattery.defect=false & oic.r.ias.zone.zonebattery.charge=100.if zonestatus=xxxx1xxxxxxx, oic.r.ias.zone.zonepowersource.powerSources =['Internal Battery'] & oic.r.ias.zone.zonebattery.defect=true & oic.r.ias.zone.zonebattery.charge=100.	
currentzonesensitivitylevel	oic.r.iaszone	ocf.currentzonesensitivitylevel	= N/A

903 Table 18 provides the details of the Properties that are part of "zcl.iaszone.info".

904 **Table 18 – The Properties of "zcl.iaszone.info".**

Zigbee Property name	Type	Required	Description
zoneID	integer	no	Unique id allocated by IAS CIE
numberofzonesensitivitylevelsupported	integer	no	Total number of sensitivity levels supported by the IAS Zone
zonestate	boolean	yes	Enrollment status of IAS Zone false=not enrolled, true=enrolled
IAS_CIE_address	string	no	Address of IAS Control and Indicating Equipment (CIE)

zonetype	string	no	Zonetype and Meaning of Alarm1 and Alarm2 zonestatus
zonestatus	array	no	x is a variable. zonestatus in Zigbee maps to zonestatus, zonebattery, and zonepowersource in OCF. Data type of zonestatus in Zigbee is 16 bitmap (xxxxxxxxxxxxxxxx) : bit 0 = Alarm1, bit 1 = Alarm2, bit 2 = Tamper, bit 3 = Battery, bit 4 = Supervision reports, bit 5 = Restore reports, bit 6 = Trouble, bit 7 = AC (mains), bit 8 = Test, bit 9 = Battery Defect. Alarm1 : 1 = opened or alarmed 0 = closed or not alarmed, Alarm2 : 1 = opened or alarmed 0 = closed or not alarmed, Tamper : 1 = Tampered 0 = Not tampered, Battery : 1 = Low battery 0 = Battery OK, Supervision reports : 1 = Reports 0 = Does not report, Restore reports : 1 = Reports restore 0 = Does not report restore, Trouble : 1 = Trouble/Failure 0 = OK, AC (mains) : 1 = AC/Mains fault 0 = AC/Mains OK, Test : 1 = Sensor is in test mode 0 = Sensor is in operation mode, Battery Defect : 1 = Sensor detects a defective battery 0 = Sensor battery is functioning.
currentzonesensitivitylevel	integer	no	Sensitivity level of IAS Zone



### 905 8.10.3 Derived model definition

```
906 {
907   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.iaszone.info.json#",
908   "$schema": "http://json-schema.org/draft-04/schema#",
909   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
910   "title": "IAS Zone Cluster - Information",
911   "definitions": {
912     "zcl.iaszone.info": {
913       "type": "object",
914       "properties": {
915         "zonestate": {
916           "type": "boolean",
917           "description": "Enrollment status of IAS Zone false=not enrolled, true=enrolled",
918           "x-ocf-conversion": {
919             "x-ocf-alias": "oic.r.iaszone",
920             "x-to-ocf": [
921               "if zonestate=0x00, ocf.zonestate=false",
922               "if zonestate=0x01, ocf.zonestate=true"
923             ],
924             "x-from-ocf": [
925               "N/A"
926             ]
927           }
928         },
929         "zonetype": {
930           "type": "string",
931           "description": "Zonetype and Meaning of Alarm1 and Alarm2 zonestatus",
932           "x-ocf-conversion": {
933             "x-ocf-alias": "oic.r.iaszone",
934             "x-to-ocf": [
935               "if zonetype=0x0000, ocf.zonetype=Standard CIE",
936               "if zonetype=0x000d, ocf.zonetype=Motion sensor",
937               "if zonetype=0x0015, ocf.zonetype=Contact switch",
938               "if zonetype=0x0028, ocf.zonetype=Fire sensor",
939               "if zonetype=0x002a, ocf.zonetype=Water sensor",
940               "if zonetype=0x002b, ocf.zonetype=Carbon Monoxide (CO) sensor",
941               "if zonetype=0x002c, ocf.zonetype=Personal emergency device",
942               "if zonetype=0x002d, ocf.zonetype=Vibration/Movement sensor",
943               "if zonetype=0x010f, ocf.zonetype=Remote Control",
944               "if zonetype=0x0115, ocf.zonetype=Key fob",
945               "if zonetype=0x021d, ocf.zonetype=Keypad",
946               "if zonetype=0x0225, ocf.zonetype=Standard Warning Device",
947               "if zonetype=0x0226, ocf.zonetype=Glass break sensor",
948               "if zonetype=0x0229, ocf.zonetype=Security repeater",
949               "if zonetype=0xffff, ocf.zonetype=Invalid Zone Type"
950             ],
951             "x-from-ocf": [
952               "N/A"
953             ]
954           }
955         },
956         "zonestatus": {
957           "type": "array",
958           "items": {
959             "type": "integer"
960           },
961           "description": "x is a variable. zonestatus in Zigbee maps to zonestatus, zonebattery,
962 and zonepowersource in OCF. Data type of zonestatus in Zigbee is 16 bitmap (xxxxxxxxxxxxxxxx) : bit
963 0 = Alarm1, bit 1 = Alarm2, bit 2 = Tamper, bit 3 = Battery, bit 4 = Supervision reports, bit 5 =
964 Restore reports, bit 6 = Trouble, bit 7 = AC (mains), bit 8 = Test, bit 9 = Battery Defect.
965 Alarm1 : 1 = opened or alarmed 0 = closed or not alarmed, Alarm2 : 1 = opened or alarmed 0 = closed
966 or not alarmed, Tamper : 1 = Tampered 0 = Not tampered, Battery : 1 = Low battery 0 = Battery OK,
967 Supervision reports : 1 = Reports 0 = Does not report, Restore reports : 1 = Reports restore 0 =
968 Does not report restore, Trouble : 1 = Trouble/Failure 0 = OK, AC (mains) : 1 = AC/Mains fault 0 =
969 AC/Mains OK, Test : 1 = Sensor is in test mode 0 = Sensor is in operation mode, Battery Defect : 1
970 = Sensor detects a defective battery 0 = Sensor battery is functioning.",
971           "x-ocf-conversion": {
972             "x-ocf-alias": "oic.r.iaszone",
973             "x-to-ocf": [
974
```

```

975         "if zonetype=0x0000 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
976         "if zonetype=0x0000 & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['system']",
977         "if zonetype=0x0000 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
978         "if zonetype=0x0000 & zonestatus=xxxxxxxxxxxxlx, ocf.zonestatus.alarms=[' ']",
979
980         "if zonetype=0x000d & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
981         "if zonetype=0x000d & zonestatus=xxxxxxxxxxxxx1,
982 ocf.zonestatus.alarms=['intrusion']",
983         "if zonetype=0x000d & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
984         "if zonetype=0x000d & zonestatus=xxxxxxxxxxxxlx,
985 ocf.zonestatus.alarms=['presence']",
986         "if zonetype=0x000d & zonestatus=xxxxxxxxxxxxl1,
987 ocf.zonestatus.alarms=['intrusion','presence']",
988
989         "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
990         "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxxx1,
991 ocf.zonestatus.alarms=['1stportalopenclose']",
992         "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
993         "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxxlx,
994 ocf.zonestatus.alarms=['2ndportalopenclose']",
995         "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxxl1,
996 ocf.zonestatus.alarms=['1stportalopenclose','2ndportalopenclose']",
997
998         "if zonetype=0x0028 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
999         "if zonetype=0x0028 & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['fire']",
1000        "if zonetype=0x0028 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1001        "if zonetype=0x0028 & zonestatus=xxxxxxxxxxxxlx, ocf.zonestatus.alarms=[' ']",
1002
1003        "if zonetype=0x002a & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1004        "if zonetype=0x002a & zonestatus=xxxxxxxxxxxxx1,
1005 ocf.zonestatus.alarms=['wateroverflow']",
1006        "if zonetype=0x002a & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1007        "if zonetype=0x002a & zonestatus=xxxxxxxxxxxxlx, ocf.zonestatus.alarms=[' ']",
1008
1009        "if zonetype=0x002b & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1010        "if zonetype=0x002b & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['CO']",
1011        "if zonetype=0x002b & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1012        "if zonetype=0x002b & zonestatus=xxxxxxxxxxxxlx,
1013 ocf.zonestatus.alarms=['cooking']",
1014        "if zonetype=0x002b & zonestatus=xxxxxxxxxxxxl1,
1015 ocf.zonestatus.alarms=['CO','cooking']",
1016
1017        "if zonetype=0x002c & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1018        "if zonetype=0x002c & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['fall']",
1019        "if zonetype=0x002c & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1020        "if zonetype=0x002c & zonestatus=xxxxxxxxxxxxlx,
1021 ocf.zonestatus.alarms=['emergencybutton']",
1022        "if zonetype=0x002c & zonestatus=xxxxxxxxxxxxl1,
1023 ocf.zonestatus.alarms=['fall','emergencybutton']",
1024
1025        "if zonetype=0x002d & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1026        "if zonetype=0x002d & zonestatus=xxxxxxxxxxxxx1,
1027 ocf.zonestatus.alarms=['movement']",
1028        "if zonetype=0x002d & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1029        "if zonetype=0x002d & zonestatus=xxxxxxxxxxxxlx,
1030 ocf.zonestatus.alarms=['vibration']",
1031        "if zonetype=0x002d & zonestatus=xxxxxxxxxxxxl1,
1032 ocf.zonestatus.alarms=['movement','vibration']",
1033
1034        "if zonetype=0x010f & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1035        "if zonetype=0x010f & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['panic']",
1036        "if zonetype=0x010f & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1037        "if zonetype=0x010f & zonestatus=xxxxxxxxxxxxlx,
1038 ocf.zonestatus.alarms=['emergency']",
1039        "if zonetype=0x010f & zonestatus=xxxxxxxxxxxxl1,
1040 ocf.zonestatus.alarms=['panic','emergency']",
1041
1042        "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1043        "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['panic']",
1044        "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1045        "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxxlx,

```

```

1046 ocf.zonestatus.alarms=['emergency']",
1047     "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxx11,
1048 ocf.zonestatus.alarms=['panic','emergency']",
1049
1050     "if zonetype=0x021d & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
1051     "if zonetype=0x021d & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['panic']",
1052     "if zonetype=0x021d & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
1053     "if zonetype=0x021d & zonestatus=xxxxxxxxxxxx1x,
1054 ocf.zonestatus.alarms=['emergency']",
1055     "if zonetype=0x021d & zonestatus=xxxxxxxxxxxx11,
1056 ocf.zonestatus.alarms=['panic','emergency']",
1057
1058     "if zonetype=0x0225 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
1059     "if zonetype=0x0225 & zonestatus=xxxxxxxxxxxx1,
1060 ocf.zonestatus.alarms=['glassbreak']",
1061     "if zonetype=0x0225 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
1062     "if zonetype=0x0225 & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['']",
1063
1064     "if zonetype=0x0226 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
1065     "if zonetype=0x0226 & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['']",
1066     "if zonetype=0x0226 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
1067     "if zonetype=0x0226 & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['']",
1068
1069     "if zonetype=0x0229 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
1070     "if zonetype=0x0229 & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['']",
1071     "if zonetype=0x0229 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
1072     "if zonetype=0x0229 & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['']",
1073
1074     "if zonetype=0xffff & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
1075     "if zonetype=0xffff & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['']",
1076     "if zonetype=0xffff & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
1077     "if zonetype=0xffff & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['']",
1078
1079     "if zonestatus=xxxxxxxx0xx, ocf.zonestatus.tamper=false",
1080     "if zonestatus=xxxxxxxx1xx, ocf.zonestatus.tamper=true",
1081
1082     "if zonestatus=xxxxxxxx0xxx, ocf.zonebattery.charge=100 &
1083 ocf.zonebattery.lowbattery=false",
1084     "if zonestatus=xxxxxxxx1xxx, ocf.zonebattery.charge=100 &
1085 ocf.zonebattery.lowbattery=true",
1086
1087     "if zonestatus=xxxxxxx0xxxx, ocf.zonestatus.zonestatusreports='none'",
1088     "if zonestatus=xxxxxxx0lxxxx, ocf.zonestatus.zonestatusreports='statuschangeonly'
1089 ",
1090     "if zonestatus=xxxxxxx10xxxx, ocf.zonestatus.zonestatusreports='alarmclearonly' ",
1091     "if zonestatus=xxxxxxx1lxxxx,
1092 ocf.zonestatus.zonestatusreports='statuschangeandalarmclear'",
1093
1094     "if zonestatus=xxxxxxx0xxxxxx, ocf.zonestatus.fault=false",
1095     "if zonestatus=xxxxxxx1xxxxxx, ocf.zonestatus.fault=true",
1096
1097     "if zonestatus=xxxxxx0xxxxxx, ocf.zonepowersource.powerSources=['AC (Mains)
1098 Power'] & ocf.zonepowersource.sourcefault=false",
1099     "if zonestatus=xxxxxx1xxxxxx, ocf.zonepowersource.powerSources=['AC (Mains)
1100 Power'] & ocf.zonepowersource.sourcefault=true",
1101
1102     "if zonestatus=xxxxx0xxxxxxx, ocf.zonestatus.test=false",
1103     "if zonestatus=xxxxx1xxxxxxx, ocf.zonestatus.test=true",
1104
1105     "if zonestatus=xxxxx0xxxxxxx, ocf.zonepowersource.powerSources=['Internal
1106 Battery'] & oic.r.ias.zone.zonebattery.defect=false & oic.r.ias.zone.zonebattery.charge=100.",
1107     "if zonestatus=xxxxx1xxxxxxx,
1108 oic.r.ias.zone.zonepowersource.powerSources=['Internal Battery'] &
1109 oic.r.ias.zone.zonebattery.defect=true & oic.r.ias.zone.zonebattery.charge=100."
1110 ],
1111 "x-from-ocf": [
1112     "N/A"
1113 ]
1114 },
1115 },
1116 "IAS_CIE_address": {

```

```

1117         "type": "string",
1118         "description": "Address of IAS Control and Indicating Equipment (CIE)",
1119         "x-ocf-conversion": {
1120             "x-ocf-alias": "oic.r.iaszone",
1121             "x-to-ocf": [
1122                 "ocf.iascieaddress= IAS_CIE_address"
1123             ],
1124             "x-from-ocf": [
1125                 "N/A"
1126             ]
1127         },
1128     },
1129     "zoneID": {
1130         "type": "integer",
1131         "description": "Unique id allocated by IAS CIE",
1132         "x-ocf-conversion": {
1133             "x-ocf-alias": "oic.r.iaszone",
1134             "x-to-ocf": [
1135                 "ocf.zoneid=zoneID"
1136             ],
1137             "x-from-ocf": [
1138                 "N/A"
1139             ]
1140         },
1141     },
1142     "numberofzonesensitivitylevelsupported": {
1143         "type": "integer",
1144         "description": "Total number of sensitivity levels supported by the IAS Zone",
1145         "x-ocf-conversion": {
1146             "x-ocf-alias": "oic.r.iaszone",
1147             "x-to-ocf": [
1148                 "ocf.numzonesensitivitylevel= numberofzonesensitivitylevelsupported"
1149             ],
1150             "x-from-ocf": [
1151                 "N/A"
1152             ]
1153         },
1154     },
1155     "currentzonesensitivitylevel": {
1156         "type": "integer",
1157         "description": "Sensitivity level of IAS Zone",
1158         "x-ocf-conversion": {
1159             "x-ocf-alias": "oic.r.iaszone",
1160             "x-to-ocf": [
1161                 "ocf.currentzonesensitivitylevel = currentzonesensitivitylevel"
1162             ],
1163             "x-from-ocf": [
1164                 "N/A"
1165             ]
1166         },
1167     },
1168 },
1169 },
1170 },
1171 "type": "object",
1172 "allOf": [
1173     {"$ref": "#/definitions/zcl.iaszone.info"}
1174 ],
1175 "required": [ "zonestate" ]
1176 }
1177

```

## 1178 8.11 Level Control Cluster - Control

### 1179 8.11.1 Derived model

1180 The derived model: "zcl.levelcontrol.control.moveto".

### 1181 8.11.2 Property definition

1182 Table 19 provides the detailed per Property mapping for "zcl.levelcontrol.control.moveto".

1183

**Table 19 – The Property mapping for "zcl.levelcontrol.control.moveto".**

Zigbee Property name	OCF Resource	To OCF	From OCF
level	oic.r.light.dimming	N/A	level=ocf.dimmingSetting * 254 /100 , transitiontime=0zcl.command.levelcontrol::movetolevel(level,transitiontime)

1184 Table 20 provides the details of the Properties that are part of "zcl.levelcontrol.control.moveto".

**Table 20 – The Properties of "zcl.levelcontrol.control.moveto".**

Zigbee name	Property	Type	Required	Description
level		integer	no	Move to certain dimming value as fast as possible

1186 **8.11.3 Derived model definition**

```

1187 {
1188   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.levelcontrol.control.json#",
1189   "$schema": "http://json-schema.org/draft-04/schema#",
1190   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1191   "title": "Level Control Cluster - Control",
1192   "definitions": {
1193     "zcl.levelcontrol.control.moveto": {
1194       "properties": {
1195         "level": {
1196           "type": "integer",
1197           "description": "Move to certain dimming value as fast as possible ",
1198           "x-ocf-conversion": {
1199             "x-ocf-alias": "oic.r.light.dimming",
1200             "x-from-ocf": [
1201               "level=ocf.dimmingSetting * 254 /100 , transitiontime=0",
1202               "zcl.command.levelcontrol::movetolevel(level,transitiontime)"
1203             ],
1204             "x-to-ocf": [
1205               "N/A"
1206             ]
1207           }
1208         }
1209       }
1210     },
1211   },
1212   "type": "object",
1213   "allOf": [
1214     {"$ref": "#/definitions/zcl.levelcontrol.control.movetolevel"}
1215   ]
1216 }
1217

```

1218 **8.12 Level Control Cluster - Information**

1219 **8.12.1 Derived model**

1220 The derived model: "zcl.levelcontrol.info".

1221 **8.12.2 Property definition**

1222 Table 21 provides the detailed per Property mapping for "zcl.levelcontrol.info".

1223

**Table 21 – The Property mapping for "zcl.levelcontrol.info".**

Zigbee name	Property	OCF Resource	To OCF	From OCF
currentlevel		oic.r.light.dimming	ocf.dimmingsetting = currentlevel/254 * 100	N/A

1224

Table 22 provides the details of the Properties that are part of "zcl.levelcontrol.info".

1225

**Table 22 – The Properties of "zcl.levelcontrol.info".**

Zigbee name	Property	Type	Required	Description
currentlevel		integer	yes	current dimming value

1226

### 8.12.3 Derived model definition

1227

1228

1229

1230

1231

1232

1233

1234

1235

1236

1237

1238

1239

1240

1241

1242

1243

1244

1245

1246

1247

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

```

{
  "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.levelcontrol.info.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Level Control Cluster - Information",
  "definitions": {
    "zcl.levelcontrol.info": {
      "type": "object",
      "properties": {
        "currentlevel": {
          "type": "integer",
          "description": "current dimming value",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.light.dimming",
            "x-to-ocf": [
              "ocf.dimmingsetting = currentlevel/254 * 100"
            ],
            "x-from-ocf": [
              "N/A"
            ]
          }
        }
      }
    }
  },
  "type": "object",
  "allOf": [
    {"$ref": "#/definitions/zcl.levelcontrol.info"}
  ],
  "required": [ "currentlevel" ]
}

```

1259

### 8.13 Occupancy Sensing Cluster - Information

1260

#### 8.13.1 Derived model

1261

The derived model: "zcl.occupancysensing.info".

1262

#### 8.13.2 Property definition

1263

Table 23 provides the detailed per Property mapping for "zcl.occupancysensing.info".

1264

**Table 23 – The Property mapping for "zcl.occupancysensing.info".**

Zigbee name	Property	OCF Resource	To OCF	From OCF
occupancy		oic.r.sensor.presence	if occupancy =xxxxxxx0, then ocf.value = falseif	N/A

		occupancy =xxxxxxx1, then ocf.value = true	
--	--	--	--

1265 Table 24 provides the details of the Properties that are part of "zcl.occupancysensing.info".

1266 **Table 24 – The Properties of "zcl.occupancysensing.info".**

Zigbee name	Property	Type	Required	Description
	occupancy	number	yes	x is a variable. Data type of occupancy in Zigbee is 8 bitmap (xxxxxxx) while data type of value in OCF is boolean type i.e., true=occupied, false=unoccupied

1267 **8.13.3 Derived model definition**

```

1268 {
1269   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.occupancysensing.info.json#",
1270   "$schema": "http://json-schema.org/draft-04/schema#",
1271   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1272   "title": "Occupancy Sensing Cluster - Information",
1273   "definitions": {
1274     "zcl.occupancysensing.info": {
1275       "type": "object",
1276       "properties": {
1277         "occupancy": {
1278           "type": "number",
1279           "description": "x is a variable. Data type of occupancy in Zigbee is 8 bitmap (xxxxxxx)
1280 while data type of value in OCF is boolean type i.e., true=occupied, false=unoccupied",
1281           "x-ocf-conversion": {
1282             "x-ocf-alias": "oic.r.sensor.presence",
1283             "x-to-ocf": [
1284               "if occupancy =xxxxxxx0, then ocf.value = false",
1285               "if occupancy =xxxxxxx1, then ocf.value = true"
1286             ],
1287             "x-from-ocf": [
1288               "N/A"
1289             ]
1290           }
1291         }
1292       }
1293     }
1294   },
1295   "type": "object",
1296   "allOf": [
1297     {"$ref": "#/definitions/zcl.occupancysensing.info"}
1298   ],
1299   "required": [ "occupancy" ]
1300 }
1301

```

1302 **8.14 On/Off Cluster - Control**

1303 **8.14.1 Derived model**

1304 The derived model: "zcl.onoff.control.off".

1305 The derived model: "zcl.onoff.control.on".

1306 **8.14.2 Property definition**

1307 Table 25 provides the detailed per Property mapping for "zcl.onoff.control.off".

1308

**Table 25 – The Property mapping for "zcl.onoff.control.off".**

Zigbee name	Property	OCF Resource	To OCF	From OCF
onoff		oic.r.switch.binary	N/A	if ocf.value = false, zcl.command.onoff::off().

1309

Table 26 provides the details of the Properties that are part of "zcl.onoff.control.off".

1310

**Table 26 – The Properties of "zcl.onoff.control.off".**

Zigbee name	Property	Type	Required	Description
onoff		boolean	no	Turn off the device

1311

Table 27 provides the detailed per Property mapping for "zcl.onoff.control.on".

1312

**Table 27 – The Property mapping for "zcl.onoff.control.on".**

Zigbee name	Property	OCF Resource	To OCF	From OCF
onoff		oic.r.switch.binary	N/A	if ocf.value = true, zcl.command.onoff::on().

1313

Table 28 provides the details of the Properties that are part of "zcl.onoff.control.on".

1314

**Table 28 – The Properties of "zcl.onoff.control.on".**

Zigbee name	Property	Type	Required	Description
onoff		boolean	no	Turn on the device

1315

**8.14.3 Derived model definition**

1316

```

{
  "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.onoff.control.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "On/Off Cluster - Control",
  "definitions": {
    "zcl.onoff.control.on": {
      "properties": {
        "onoff": {
          "type": "boolean",
          "description": "Turn on the device",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.switch.binary",
            "x-from-ocf": [
              "if ocf.value = true, zcl.command.onoff::on()."
            ],
            "x-to-ocf": [
              "N/A"
            ]
          }
        }
      }
    }
  },
  "zcl.onoff.control.off": {
    "properties": {
      "onoff": {
        "type": "boolean",
        "description": "Turn off the device",
        "x-ocf-conversion": {
          "x-ocf-alias": "oic.r.switch.binary",
          "x-from-ocf": [
            "if ocf.value = false, zcl.command.onoff::off()."
          ]
        }
      }
    }
  }
}

```

1347



```

1348         ],
1349         "x-to-ocf": [
1350             "N/A"
1351         ]
1352     }
1353 }
1354 }
1355 }
1356 },
1357 "type": "object",
1358 "allOf": [
1359     {"$ref": "#/definitions/zcl.onoff.control.on"},
1360     {"$ref": "#/definitions/zcl.onoff.control.off"}
1361 ]
1362 }
1363

```

1364 **8.15 On/off Cluster - Information**

1365 **8.15.1 Derived model**

1366 The derived model: "zcl.onoff".

1367 **8.15.2 Property definition**

1368 Table 29 provides the detailed per Property mapping for "zcl.onoff".

1369 **Table 29 – The Property mapping for "zcl.onoff".**

Zigbee name	Property	OCF Resource	To OCF	From OCF
onoff		oic.r.switch.binary	if onoff = false, then ocf.value = false if onoff = true, then ocf.value = true	N/A

1370 Table 30 provides the details of the Properties that are part of "zcl.onoff".

1371 **Table 30 – The Properties of "zcl.onoff".**

Zigbee name	Property	Type	Required	Description
onoff		boolean	yes	On/off status of the device

1372 **8.15.3 Derived model definition**

```

1373 {
1374     "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.onoff.info.json#",
1375     "$schema": "http://json-schema.org/draft-04/schema#",
1376     "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1377     "title": "On/off Cluster - Information",
1378     "definitions": {
1379         "zcl.onoff": {
1380             "type": "object",
1381             "properties": {
1382                 "onoff": {
1383                     "type": "boolean",
1384                     "description": "On/off status of the device",
1385                     "x-ocf-conversion": {
1386                         "x-ocf-alias": "oic.r.switch.binary",
1387                         "x-to-ocf": [
1388                             "if onoff = false, then ocf.value = false",
1389                             "if onoff = true, then ocf.value = true"
1390                         ],
1391                         "x-from-ocf": [
1392                             "N/A"
1393                         ]
1394                     }
1395                 }
1396             }
1397         }
1398     }
1399 }

```

```

1394     }
1395   }
1396 }
1397 }
1398 },
1399 "type": "object",
1400 "allOf": [
1401   {"$ref": "#/definitions/zcl.onoff.info"}
1402 ],
1403 "required": [ "onoff" ]
1404 }
1405

```

1406 **8.16 Temperature Measurement Cluster - Information**

1407 **8.16.1 Derived model**

1408 The derived model: "zcl.temperaturemeasurement.info".

1409 **8.16.2 Property definition**

1410 Table 31 provides the detailed per Property mapping for "zcl.temperaturemeasurement.info".

1411 **Table 31 – The Property mapping for "zcl.temperaturemeasurement.info".**

Zigbee name	Property	OCF Resource	To OCF	From OCF
MeasuredValue		oic.r.temperature	ocf.temperature = MeasuredValue/100units = C	N/A
MinMeasuredValue		oic.r.temperature	ocf.range[0] = MinMeasuredValue/100	N/A
Tolerance		oic.r.temperature	ocf.precision = Tolerance/100	N/A
MaxMeasuredValue		oic.r.temperature	ocf.range[1] = MaxMeasuredValue/100	N/A

1412 Table 32 provides the details of the Properties that are part of "zcl.temperaturemeasurement.info".

1413 **Table 32 – The Properties of "zcl.temperaturemeasurement.info".**

Zigbee name	Property	Type	Required	Description
MeasuredValue		number	yes	Measured value
MinMeasuredValue		number	yes	Minimum value of MeasuredValue
Tolerance		number	yes	Magnitude of the possible error
MaxMeasuredValue		number	yes	Maximum value of MeasuredValue

1414 **8.16.3 Derived model definition**

```

1415 {
1416   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.temperaturemeasurement.info.json#",
1417   "$schema": "http://json-schema.org/draft-04/schema#",
1418   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1419   "title": "Temperature Measurement Cluster - Information",
1420   "definitions": {
1421     "zcl.temperaturemeasurement.info": {
1422       "type": "object",
1423       "properties": {
1424         "MeasuredValue": {
1425           "type": "number",
1426           "description": "Measured value",

```

```

1427     "x-ocf-conversion": {
1428       "x-ocf-alias": "oic.r.temperature",
1429       "x-to-ocf": [
1430         "ocf.temperature = MeasuredValue/100",
1431         "units = C"
1432       ],
1433       "x-from-ocf": [
1434         "N/A"
1435       ]
1436     },
1437   },
1438   "Tolerance": {
1439     "type": "number",
1440     "description": "Magnitude of the possible error",
1441     "x-ocf-conversion": {
1442       "x-ocf-alias": "oic.r.temperature",
1443       "x-to-ocf": [
1444         "ocf.precision = Tolerance/100"
1445       ],
1446       "x-from-ocf": [
1447         "N/A"
1448       ]
1449     },
1450   },
1451   "MinMeasuredValue": {
1452     "type": "number",
1453     "description": "Minimum value of MeasuredValue",
1454     "x-ocf-conversion": {
1455       "x-ocf-alias": "oic.r.temperature",
1456       "x-to-ocf": [
1457         "ocf.range[0] = MinMeasuredValue/100"
1458       ],
1459       "x-from-ocf": [
1460         "N/A"
1461       ]
1462     },
1463   },
1464   "MaxMeasuredValue": {
1465     "type": "number",
1466     "description": "Maximum value of MeasuredValue",
1467     "x-ocf-conversion": {
1468       "x-ocf-alias": "oic.r.temperature",
1469       "x-to-ocf": [
1470         "ocf.range[1] = MaxMeasuredValue/100"
1471       ],
1472       "x-from-ocf": [
1473         "N/A"
1474       ]
1475     },
1476   },
1477 },
1478 },
1479 },
1480 "type": "object",
1481 "allOf": [
1482   {"$ref": "#/definitions/zcl.temperaturemeasurement.info"}
1483 ],
1484 "required": [ "MeasuredValue", "Tolerance", "MinMeasuredValue", "MaxMeasuredValue" ]
1485 }
1486

```

## 1487 8.17 Thermostat Cluster - Cool - Control

### 1488 8.17.1 Derived model

1489 The derived model: "zcl.thermostat\_cool.control.setpointraiselower".

### 1490 8.17.2 Property definition

1491 Table 33 provides the detailed per Property mapping for  
 1492 "zcl.thermostat\_cool.control.setpointraiselower".

1493 **Table 33 – The Property mapping for "zcl.thermostat\_cool.control.setpointraiselower".**

Zigbee Property name	OCF Resource	To OCF	From OCF
amount	oic.r.temperature	N/A	if ocf.temperature is updated, then amount=ocf.temperature*100.zcl.command.thermostat::setpointraiselower(mode, amount)

1494 Table 34 provides the details of the Properties that are part of  
 1495 "zcl.thermostat\_cool.control.setpointraiselower".

1496 **Table 34 – The Properties of "zcl.thermostat\_cool.control.setpointraiselower".**

Zigbee name	Property	Type	Required	Description
amount		number	no	Set the target temperature with cool mode. Mode=0x01 is set by Zigbee 3.0 translator

1497 **8.17.3 Derived model definition**

```

1498 {
1499   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.thermostat_cool.control.json#",
1500   "$schema": "http://json-schema.org/draft-04/schema#",
1501   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1502   "title": "Thermostat Cluster - Cool - Control",
1503   "definitions": {
1504     "zcl.thermostat_cool.control.setpointraiselower": {
1505       "type": "object",
1506       "properties": {
1507         "amount": {
1508           "type": "number",
1509           "description": "Set the target temperature with cool mode. Mode=0x01 is set by Zigbee 3.0
1510 translator",
1511           "x-ocf-conversion": {
1512             "x-ocf-alias": "oic.r.temperature",
1513             "x-from-ocf": [
1514               "if ocf.temperature is updated, then amount= ocf.temperature*100.",
1515               "zcl.command.thermostat::setpointraiselower(mode, amount)"
1516             ],
1517             "x-to-ocf": [
1518               "N/A"
1519             ]
1520           }
1521         }
1522       }
1523     },
1524     "type": "object",
1525     "allof": [
1526       {"$ref": "#/definitions/zcl.thermostat_cool.control.setpointraiselower"}
1527     ]
1528   }
1529 }
1530

```

1531 **8.18 Thermostat Cluster - Current Temperature - Information**

1532 **8.18.1 Derived model**

1533 The derived model: "zcl.thermostat\_currenttemperature.info".

1534 **8.18.2 Property definition**

1535 Table 35 provides the detailed per Property mapping for "zcl.thermostat\_currenttemperature.info".

1536 **Table 35 – The Property mapping for "zcl.thermostat\_currenttemperature.info".**

Zigbee Property name	OCF Resource	To OCF	From OCF
localtemperature	oic.r.temperature	ocf.temperature=localtempearture/100units = C	N/A

1537 Table 36 provides the details of the Properties that are part of  
 1538 "zcl.thermostat\_currenttemperature.info".

1539 **Table 36 – The Properties of "zcl.thermostat\_currenttemperature.info".**

Zigbee Property name	Type	Required	Description
localtemperature	number	no	current sensed temperature

1540 **8.18.3 Derived model definition**

```

1541 {
1542   "id":
1543   "http://openinterconnect.org/zigbeemapping/schemas/zcl.thermostat_currenttemperature.info.json#",
1544   "$schema": "http://json-schema.org/draft-04/schema#",
1545   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1546   "title": "Thermostat Cluster - Current Temperature - Information ",
1547   "definitions": {
1548     "zcl.thermostat_currenttemperature.info": {
1549       "type": "object",
1550       "properties": {
1551         "localtemperature": {
1552           "type": "number",
1553           "description": "current sensed temperature",
1554           "x-ocf-conversion": {
1555             "x-ocf-alias": "oic.r.temperature",
1556             "x-to-ocf": [
1557               "ocf.temperature=localtempearture/100",
1558               "units = C"
1559             ],
1560             "x-from-ocf": [
1561               "N/A"
1562             ]
1563           }
1564         }
1565       }
1566     }
1567   },
1568   "type": "object",
1569   "allOf": [
1570     {"$ref": "#/definitions/zcl.thermostat_currenttemperature.info"}
1571   ],
1572   "required": [ "localtempearture" ]
1573 }
1574

```

1575 **8.19 Thermostat Cluster - Heat - Control**

1576 **8.19.1 Derived model**

1577 The derived model: "zcl.thermostat\_heat.control.setpointraiseLower".

1578 **8.19.2 Property definition**

1579 Table 37 provides the detailed per Property mapping for  
 1580 "zcl.thermostat\_heat.control.setpointraiseLower".

1581 **Table 37 – The Property mapping for "zcl.thermostat\_heat.control.setpointraiseLower".**

Zigbee Property name	OCF Resource	To OCF	From OCF
amount	oic.r.temperature	N/A	if ocf.temperature is updated, then amount= ocf.temperature*100.zcl.command.thermostat::setpointraiseLower(mode, amount)

1582 Table 38 provides the details of the Properties that are part of  
 1583 "zcl.thermostat\_heat.control.setpointraiseLower".

1584 **Table 38 – The Properties of "zcl.thermostat\_heat.control.setpointraiseLower".**

Zigbee name	Property	Type	Required	Description
amount		number	no	Set the target temperature with heat mode. Mode=0x00 is set by Zigbee 3.0 translator

1585 **8.19.3 Derived model definition**

```

1586 {
1587   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.thermostat_heat.control.json#",
1588   "$schema": "http://json-schema.org/draft-04/schema#",
1589   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1590   "title": "Thermostat Cluster - Heat - Control",
1591   "definitions": {
1592     "zcl.thermostat_heat.control.setpointraiseLower": {
1593       "type": "object",
1594       "properties": {
1595         "amount": {
1596           "type": "number",
1597           "description": "Set the target temperature with heat mode. Mode=0x00 is set by Zigbee 3.0
1598 translator",
1599           "x-ocf-conversion": {
1600             "x-ocf-alias": "oic.r.temperature",
1601             "x-from-ocf": [
1602               "if ocf.temperature is updated, then amount= ocf.temperature*100.",
1603               "zcl.command.thermostat::setpointraiseLower(mode, amount)"
1604             ],
1605             "x-to-ocf": [
1606               "N/A"
1607             ]
1608           }
1609         }
1610       }
1611     }
1612   },
1613   "type": "object",
1614   "allof": [
1615     {"$ref": "#/definitions/zcl.thermostat_heat.control.setpointraiseLower"}
1616   ]
1617 }
1618
  
```

1619 **8.20 Window Covering Cluster - Configuration - Control**

1620 **8.20.1 Derived model**

1621 The derived model: "zcl.windowcovering\_conf.control".

1622 **8.20.2 Property definition**

1623 Table 39 provides the detailed per Property mapping for "zcl.windowcovering\_conf.control".

1624 **Table 39 – The Property mapping for "zcl.windowcovering\_conf.control".**

Zigbee Property name	OCF Resource	To OCF	From OCF
Acceleration Time-Lift	oic.r.windowcovering	N/A	if ocf.liftaccelerationtime is updated, Acceleration Time-Lift=ocf.liftaccelerationtime.zcl.command.general::write(Acceleration Time-Lift)
Velocity-Lift	oic.r.windowcovering	N/A	if ocf.liftvelocity is updated, Velocity-Lift = ocf.liftvelocity.zcl.command.general::write(Velocity-Lift)
Deceleration Time-Lift	oic.r.windowcovering	N/A	if ocf.liftdecelerationtime is updated, Deceleration Time-Lift=ocf.liftdecelerationtime.zcl.command.general::write(Deceleration Time-Lift)
mode	oic.r.windowcovering	N/A	if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxxxxx0.if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxxxxx1.if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxxxx0x.if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxxxx1x.if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxxx0xx.if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxxx1xx.if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxx0xxx.if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxx1xxx.zcl.command.general::write(mode)

1625 Table 40 provides the details of the Properties that are part of "zcl.windowcovering\_conf.control".

1626 **Table 40 – The Properties of "zcl.windowcovering\_conf.control".**

Zigbee Property name	Type	Required	Description
Acceleration Time-Lift	integer	no	Set ramp up times to reaching the velocity setting (0.1sec).
Velocity-Lift	integer	no	Set velocity associated with Lifting the Window Covering (cm/sec).
Deceleration Time-Lift	integer	no	Set ramp down times associated with stoping the velocity setting (0.1sec).
mode	integer	no	Set the mode. x is a variable. Data type of Mode in Zigbee is 8 bitmap (xxxxxxx) while data type of mode in OCF is array with 4 Boolean type items(i.e., [Reversed Motor Direction, Calibration Mode, Maintenance Mode,

			LED]). Reversed Motor Direction : 0 = motor direction is normal, 1 = motor direction is reversed. Calibration Mode : 0 = run in normal mode, 1 = run in calibration mode. Maintenance Mode : 0 = motor is running normally, 1 = motor is running in maintenance mode. LED: 0 = LEDs are off, 1 = LEDs will display feedback.
--	--	--	--

1627 **8.20.3 Derived model definition**

```

1628 {
1629   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_conf.control.json#",
1630   "$schema": "http://json-schema.org/draft-04/schema#",
1631   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1632   "title": "Window Covering Cluster - Configuration - Control",
1633   "definitions": {
1634     "zcl.windowcovering_conf.control": {
1635       "properties": {
1636         "mode": {
1637           "type": "integer",
1638           "description": "Set the mode. x is a variable. Data type of Mode in Zigbee is 8 bitmap
1639 (xxxxxxx) while data type of mode in OCF is array with 4 Boolean type items(i.e., [Reversed Motor
1640 Direction, Calibration Mode, Maintenance Mode, LED]). Reversed Motor Direction : 0 = motor
1641 direction is normal, 1 = motor direction is reversed. Calibration Mode : 0 = run in normal mode, 1
1642 = run in calibration mode. Maintenance Mode : 0 = motor is running normally, 1 = motor is running
1643 in maintenance mode. LED: 0 = LEDs are off, 1 = LEDs will display feedback.",
1644           "x-ocf-conversion": {
1645             "x-ocf-alias": "oic.r.windowcovering",
1646             "x-from-ocf": [
1647               "if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxxxxx0.",
1648               "if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxxxxx1.",
1649               "if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxxxxx0x.",
1650               "if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxxxxx1x.",
1651               "if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxxxx0xx.",
1652               "if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxxxx1xx.",
1653               "if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxxx0xxx.",
1654               "if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxxx1xxx.",
1655               "zcl.command.general::write(mode)"
1656             ],
1657             "x-to-ocf": [
1658               "N/A"
1659             ]
1660           }
1661         },
1662         "Velocity-Lift": {
1663           "type": "integer",
1664           "description": "Set velocity associated with Lifting the Window Covering (cm/sec).",
1665           "x-ocf-conversion": {
1666             "x-ocf-alias": "oic.r.windowcovering",
1667             "x-from-ocf": [
1668               "if ocf.liftvelocity is updated, Velocity-Lift = ocf.liftvelocity.",
1669               "zcl.command.general::write(Velocity-Lift)"
1670             ],
1671             "x-to-ocf": [
1672               "N/A"
1673             ]
1674           }
1675         }
1676       }
1677     }
1678   }

```



```

1676     "Acceleration Time-Lift": {
1677         "type": "integer",
1678         "description": "Set ramp up times to reaching the velocity setting (0.1sec).",
1679         "x-ocf-conversion": {
1680             "x-ocf-alias": "oic.r.windowcovering",
1681             "x-from-ocf": [
1682                 "if ocf.liftaccelerationtime is updated, Acceleration Time-
1683 Lift=ocf.liftaccelerationtime.",
1684                 "zcl.command.general::write(Acceleration Time-Lift)"
1685             ],
1686             "x-to-ocf": [
1687                 "N/A"
1688             ]
1689         }
1690     },
1691     "Deceleration Time-Lift": {
1692         "type": "integer",
1693         "description": "Set ramp down times associated with stoping the velocity setting
1694 (0.1sec).",
1695         "x-ocf-conversion": {
1696             "x-ocf-alias": "oic.r.windowcovering",
1697             "x-from-ocf": [
1698                 "if ocf.liftdecelerationtime is updated, Deceleration Time-
1699 Lift=ocf.liftdecelerationtime.",
1700                 "zcl.command.general::write(Deceleration Time-Lift)"
1701             ],
1702             "x-to-ocf": [
1703                 "N/A"
1704             ]
1705         }
1706     }
1707 }
1708 }
1709 },
1710 "type": "object",
1711 "allOf": [
1712     {"$ref": "#/definitions/zcl.windowcovering_conf.control"}
1713 ]
1714 }
1715

```

1716 **8.21 Window Covering Cluster - Configuration - Information**

1717 **8.21.1 Derived model**

1718 The derived model: "zcl.windowcovering\_conf.info".

1719 **8.21.2 Property definition**

1720 Table 41 provides the detailed per Property mapping for "zcl.windowcovering\_conf.info".

1721 **Table 41 – The Property mapping for "zcl.windowcovering\_conf.info".**

Zigbee name	Property	OCF Resource	To OCF	From OCF
Velocity-Lift		oic.r.windowcovering	ocf.liftvelocity = Velocity-Lift	N/A
Windowcoveringtype		oic.r.windowcovering	if WindowCoveringType=0x00, ocf.windowcoveringtype= Rollershade.if WindowCoveringType=0x01, ocf.windowcoveringtype= RollerShade-2 Motor.if WindowCoveringType=0x02, ocf.windowcoveringtype= RollerShade-Exterior.if WindowCoveringType=0x03, ocf.windowcoveringtype=	N/A

		RollerShade-Exterior-2 Motor.if WindowCoveringType=0x04, ocf.windowcoveringtype= Drapery.if WindowCoveringType=0x05, ocf.windowcoveringtype= Awning.if WindowCoveringType=0x06, ocf.windowcoveringtype= Shutter.if WindowCoveringType=0x07, ocf.windowcoveringtype= Tilt Blind - Tilt Only.if WindowCoveringType=0x08, ocf.windowcoveringtype= Tilt Blind &Lift and Tilt.if WindowCoveringType=0x09, ocf.windowcoveringtype= Projector Screen.	
Config/Status	oic.r.windowcovering	if Config/Status =xxxxxxx0, ocf.configstatus.operational = falseif Config/Status =xxxxxxx1, ocf.configstatus.operational = trueif Config/Status =xxxxxx0x, ocf.configstatus.online = falseif Config/Status =xxxxxx1x, ocf.configstatus.online = trueif Config/Status =xxxxxx0xx, ocf.configstatus.rotationdirection = 'normal'if Config/Status =xxxxxx1xx, ocf.configstatus.rotationdirection = 'reversed'if Config/Status =xxxx0xxx, ocf.configstatus.controllift = 'openloop'if Config/Status =xxxx1xxx, ocf.configstatus.controllift = 'closedloop'if Config/Status =xxx0xxxx, ocf.configstatus.controllift = 'openloop'if Config/Status =xxx1xxx, ocf.configstatus.controllift = 'closedloop'if Config/Status =xx0xxxx, ocf.configstatus.closedloopliftcontrol = 'timer'if Config/Status =xx1xxxx, ocf.configstatus.closedloopliftcontrol = 'encoder'if Config/Status =x0xxxx, ocf.configstatus.closedlooptiltcontrol = 'timer'if Config/Status =x1xxxx, ocf.configstatus.closedlooptiltcontrol = 'encoder'	N/A
Deceleration Time-Lift	oic.r.windowcovering	ocf.liftdecelerationtime=	N/A
Mode	oic.r.windowcovering	if Mode =xxxxxxx0, ocf.mode.motordirection = falseif Mode =xxxxxxx1,	N/A

		ocf.mode.motordirection = trueif Mode =xxxxxx0x, ocf.mode.calibration = falseif Mode =xxxxxx1x, ocf.mode.calibration = trueif Mode =xxxxx0xx, ocf.mode.maintenance = falseif Mode =xxxxx1xx, ocf.mode.maintenance = trueif Mode =xxxx0xxx, ocf.mode.ledfeedback = falseif Mode =xxx1xxx, ocf.mode.ledfeedback = true	
Acceleration Time-Lift	oic.r.windowcovering	ocf.liftaccelerationtime=	Acceleration Time-Lift
			N/A

1722 Table 42 provides the details of the Properties that are part of "zcl.windowcovering\_conf.info".

1723 **Table 42 – The Properties of "zcl.windowcovering\_conf.info".**

Zigbee name	Property	Type	Required	Description
Velocity-Lift		integer	no	Velocity associated with Lifting the Window Covering (cm/sec).
Windowcoveringtype		string	yes	Type of Window Covering(i.e., [Rollershade, RollerShade-2 Motor, RollerShade-Exterior, RollerShade-Exterior-2 Motor, Drapery, Awning, Shutter, Tilt Blind - Tilt Only, Tilt Blind â€™ Lift and Tilt, Projector Screen])
Config/Status		integer	yes	x is a variable. Config/Status in Zigbee maps to configstatus in OCF. Data type of Config/Status in Zigbee is 8 bitmap (xxxxxxx) : bit 0 = Operational, bit 1 = Online, bit 2 = Reversal, bit 3 = Control-Lift, bit 4 = Control-Tilt, bit 5 = Encoder-Lift, bit 6 = Encoder-Tilt. Operational: This status bit defines if the Window Covering is operational. 0 = Not Operational, 1 = Operational. Online: This status bit defines if the Window Covering is enabled for transmitting over the ZigBee network. 0 = Not Online, 1 = Online. Reversal: This status bit

			<p>identifies if the direction of rotation for the Window Covering has been reversed in order for Open/Up commands to match the physical installation condition. 0 = Commands are normal, 1 = Open/Up Commands have been reversed.</p> <p>Control Lift: This status bit identifies if the window covering supports Open Loop or Closed Loop Lift Control. 0 = Lift control is Open Loop, 1 = Lift control is Closed.</p> <p>Control Tilt: This status bit identifies if the window covering supports Open Loop or Closed Loop Tilt Control. 0 = Tilt control is Open Loop, 1 = Tilt control is Closed.</p> <p>Encoder Lift: This status bit identifies if a Closed Loop Controlled Window Covering is employing an encoder for positioning the height of the window covering. 0 = Timer Controlled, 1 = Encoder Controlled.</p> <p>Encoder Tilt: This status bit identifies if a Closed Loop Controlled Window Covering is employing an encoder for tilting the window covering. 0 = Timer Controlled, 1 = Encoder Controlled.</p>
Deceleration Time-Lift	integer	no	Ramp down times associated with stoping the velocity setting (0.1sec).
Mode	integer	yes	x is a variable. Mode in Zigbee maps to mode in OCF. Data type of Mode in Zigbee is 8 bitmap (xxxxxxx) : bit 0 = Reversed Motor Direction, bit 1 = Calibration Mode, bit 2 = Maintenance Mode, bit 3 = LED. Reversed Motor Direction : 0 = motor direction is normal, 1 = motor direction is

			reversed. Calibration Mode : 0 = run in normal mode, 1 = run in calibration mode. Maintenance Mode : 0 = motor is running normally, 1 = motor is running in maintenance mode. LED: 0 = LEDs are off, 1 = LEDs will display feedback.
Acceleration Time-Lift	integer	no	Ramp up times to reaching the velocity setting (0.1sec).

1724 **8.21.3 Derived model definition**

```

1725 {
1726   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_conf.info.json#",
1727   "$schema": "http://json-schema.org/draft-04/schema#",
1728   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1729   "title": "Window Covering Cluster - Configuration - Information",
1730   "definitions": {
1731     "zcl.windowcovering_conf.info": {
1732       "type": "object",
1733       "properties": {
1734         "Windowcoveringtype": {
1735           "type": "string",
1736           "description": "Type of Window Covering(i.e., [Rollershade,RollerShade-2 Motor,
1737 RollerShade-Exterior, RollerShade-Exterior-2 Motor, Drapery, Awning, Shutter, Tilt Blind - Tilt
1738 Only, Tilt Blind â€ Lift and Tilt, Projector Screen])",
1739           "x-ocf-conversion": {
1740             "x-ocf-alias": "oic.r.windowcovering",
1741             "x-to-ocf": [
1742               "if WindowCoveringType=0x00, ocf.windowcoveringtype= Rollershade.",
1743               "if WindowCoveringType=0x01, ocf.windowcoveringtype= RollerShade-2 Motor.",
1744               "if WindowCoveringType=0x02, ocf.windowcoveringtype= RollerShade-Exterior.",
1745               "if WindowCoveringType=0x03, ocf.windowcoveringtype= RollerShade-Exterior-2 Motor.",
1746               "if WindowCoveringType=0x04, ocf.windowcoveringtype= Drapery.",
1747               "if WindowCoveringType=0x05, ocf.windowcoveringtype= Awning.",
1748               "if WindowCoveringType=0x06, ocf.windowcoveringtype= Shutter.",
1749               "if WindowCoveringType=0x07, ocf.windowcoveringtype= Tilt Blind - Tilt Only.",
1750               "if WindowCoveringType=0x08, ocf.windowcoveringtype= Tilt Blind â€ Lift and Tilt.",
1751               "if WindowCoveringType=0x09, ocf.windowcoveringtype= Projector Screen."
1752             ],
1753             "x-from-ocf": [
1754               "N/A"
1755             ]
1756           }
1757         },
1758         "Config/Status": {
1759           "type": "integer",
1760           "description": " x is a variable. Config/Status in Zigbee maps to configstatus in OCF.
1761 Data type of Config/Status in Zigbee is 8 bitmap (xxxxxxx) : bit 0 = Operational, bit 1 = Online,
1762 bit 2 = Reversal, bit 3 = Control-Lift, bit 4 = Control-Tilt, bit 5 = Encoder-Lift, bit 6 =
1763 Encoder-Tilt. Operational: This status bit defines if the Window Covering is operational. 0 = Not
1764 Operational, 1 = Operational. Online: This status bit defines if the Window Covering is enabled for
1765 transmitting over the ZigBee network. 0 = Not Online, 1 = Online. Reversal: This status bit
1766 identifies if the direction of rotation for the Window Covering has been reversed in order for
1767 Open/Up commands to match the physical installation condition. 0 = Commands are normal, 1 = Open/Up
1768 Commands have been reversed. Control Lift: This status bit identifies if the window covering
1769 supports Open Loop or Closed Loop Lift Control. 0 = Lift control is Open Loop, 1 = Lift control is
1770 Closed. Control Tilt: This status bit identifies if the window covering supports Open Loop or
1771 Closed Loop Tilt Control. 0 = Tilt control is Open Loop, 1 = Tilt control is Closed. Encoder Lift:
1772 This status bit identifies if a Closed Loop Controlled Window Covering is employing an encoder for
1773 positioning the height of the window covering. 0 = Timer Controlled, 1 = Encoder Controlled.
1774 Encoder Tilt: This status bit identifies if a Closed Loop Controlled Window Covering is employing
1775 an encoder for tilting the window covering. 0 = Timer Controlled, 1 = Encoder Controlled.",
1776           "x-ocf-conversion": {

```

```

1777     "x-ocf-alias": "oic.r.windowcovering",
1778     "x-to-ocf": [
1779         "if Config/Status =xxxxxxx0, ocf.configstatus.operational = false",
1780         "if Config/Status =xxxxxxx1, ocf.configstatus.operational = true",
1781         "if Config/Status =xxxxxxx0x, ocf.configstatus.online = false",
1782         "if Config/Status =xxxxxxx1x, ocf.configstatus.online = true",
1783         "if Config/Status =xxxxx0xx, ocf.configstatus.rotationdirection = 'normal'",
1784         "if Config/Status =xxxxx1xx, ocf.configstatus.rotationdirection = 'reversed'",
1785         "if Config/Status =xxxx0xxx, ocf.configstatus.controllift = 'openloop'",
1786         "if Config/Status =xxxx1xxx, ocf.configstatus.controllift = 'closedloop'",
1787         "if Config/Status =xxx0xxxx, ocf.configstatus.controltilt = 'openloop'",
1788         "if Config/Status =xxx1xxxx, ocf.configstatus.controltilt = 'closedloop'",
1789         "if Config/Status =xx0xxxxx, ocf.configstatus.closedloopliftcontrol = 'timer'",
1790         "if Config/Status =xx1xxxxx, ocf.configstatus.closedloopliftcontrol = 'encoder'",
1791         "if Config/Status =x0xxxxxx, ocf.configstatus.closedlooptiltcontrol = 'timer'",
1792         "if Config/Status =x1xxxxxx, ocf.configstatus.closedlooptiltcontrol = 'encoder'"
1793     ],
1794     "x-from-ocf": [
1795         "N/A"
1796     ]
1797 },
1798 },
1799 "Mode": {
1800     "type": "integer",
1801     "description": "x is a variable. Mode in Zigbee maps to mode in OCF. Data type of Mode in
1802 Zigbee is 8 bitmap (xxxxxxx) : bit 0 = Reversed Motor Direction, bit 1 = Calibration Mode, bit 2 =
1803 Maintenance Mode, bit 3 = LED. Reversed Motor Direction : 0 = motor direction is normal, 1 = motor
1804 direction is reversed. Calibration Mode : 0 = run in normal mode, 1 = run in calibration mode.
1805 Maintenance Mode : 0 = motor is running normally, 1 = motor is running in maintenance mode. LED: 0
1806 = LEDs are off, 1 = LEDs will display feedback.",
1807     "x-ocf-conversion": {
1808         "x-ocf-alias": "oic.r.windowcovering",
1809         "x-to-ocf": [
1810             "if Mode =xxxxxxx0, ocf.mode.motordirection = false",
1811             "if Mode =xxxxxxx1, ocf.mode.motordirection = true",
1812             "if Mode =xxxxxxx0x, ocf.mode.calibration = false",
1813             "if Mode =xxxxxxx1x, ocf.mode.calibration = true",
1814             "if Mode =xxxxx0xx, ocf.mode.maintenance = false",
1815             "if Mode =xxxxx1xx, ocf.mode.maintenance = true",
1816             "if Mode =xxxx0xxx, ocf.mode.ledfeedback = false",
1817             "if Mode =xxxx1xxx, ocf.mode.ledfeedback = true"
1818         ],
1819         "x-from-ocf": [
1820             "N/A"
1821         ]
1822     }
1823 },
1824 "Velocity-Lift": {
1825     "type": "integer",
1826     "description": "Velocity associated with Lifting the Window Covering (cm/sec).",
1827     "x-ocf-conversion": {
1828         "x-ocf-alias": "oic.r.windowcovering",
1829         "x-to-ocf": [
1830             "ocf.liftvelocity = Velocity-Lift"
1831         ],
1832         "x-from-ocf": [
1833             "N/A"
1834         ]
1835     }
1836 },
1837 "Acceleration Time-Lift": {
1838     "type": "integer",
1839     "description": "Ramp up times to reaching the velocity setting (0.1sec).",
1840     "x-ocf-conversion": {
1841         "x-ocf-alias": "oic.r.windowcovering",
1842         "x-to-ocf": [
1843             "ocf.liftaccelerationtime= Acceleration Time-Lift"
1844         ],
1845         "x-from-ocf": [
1846             "N/A"
1847     ]

```

```

1848     }
1849   },
1850   "Deceleration Time-Lift": {
1851     "type": "integer",
1852     "description": "Ramp down times associated with stoping the velocity setting (0.1sec).",
1853     "x-ocf-conversion": {
1854       "x-ocf-alias": "oic.r.windowcovering",
1855       "x-to-ocf": [
1856         "ocf.liftdecelerationtime= Deceleration Time-Lift"
1857       ],
1858       "x-from-ocf": [
1859         "N/A"
1860       ]
1861     }
1862   }
1863 }
1864 }
1865 },
1866 "type": "object",
1867 "allOf": [
1868   {"$ref": "#/definitions/zcl.windowcovering_conf.info"}
1869 ],
1870 "required": [ "Windowcoveringtype", "Config/Status", "Mode" ]
1871 }
1872

```

1873 **8.22 Window Covering Cluster - Lift Percentage - Control**

1874 **8.22.1 Derived model**

1875 The derived model: "zcl.windowcovering\_liftpercentage.control.gotoliftpercentage".

1876 **8.22.2 Property definition**

1877 Table 43 provides the detailed per Property mapping for  
 1878 "zcl.windowcovering\_liftpercentage.control.gotoliftpercentage".

1879 **Table 43 – The Property mapping for**  
 1880 **"zcl.windowcovering\_liftpercentage.control.gotoliftpercentage".**

Zigbee Property name	OCF Resource	To OCF	From OCF
percentageliftvalue	oic.r.openlevel	N/A	if ocf.openLevel is updated, percentage lift value = ocf.openLevel.zcl.command.windowcovering::gotoliftpercentage(percentageliftvalue)

1881 Table 44 provides the details of the Properties that are part of  
 1882 "zcl.windowcovering\_liftpercentage.control.gotoliftpercentage".

1883 **Table 44 – The Properties of**  
 1884 **"zcl.windowcovering\_liftpercentage.control.gotoliftpercentage".**

Zigbee name	Property	Type	Required	Description
percentageliftvalue		integer	no	Adjust the window at the percentage lift value.

1885 **8.22.3 Derived model definition**

```

1886 {
1887   "id":
1888   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_liftpercentage.control.json#"
1889   ,
1890   "$schema": "http://json-schema.org/draft-04/schema#",
1891   "description" : "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",

```

```

1892     "title": "Window Covering Cluster - Lift Percentage - Control",
1893     "definitions": {
1894         "zcl.windowcovering_liftpercentage.control.gotoliftpercentage": {
1895             "properties": {
1896                 "percentageliftvalue": {
1897                     "type": "integer",
1898                     "description": "Adjust the window at the percentage lift value.",
1899                     "x-ocf-conversion": {
1900                         "x-ocf-alias": "oic.r.openlevel",
1901                         "x-from-ocf": [
1902                             "if ocf.openLevel is updated, percentage lift value = ocf.openLevel.",
1903                             "zcl.command.windowcovering::gotoliftpercentage(percentageliftvalue)"
1904                         ],
1905                         "x-to-ocf": [
1906                             "N/A"
1907                         ]
1908                     }
1909                 }
1910             }
1911         }
1912     },
1913     "type": "object",
1914     "allOf": [
1915         {"$ref": "#/definitions/zcl.windowcovering_liftpercentage.control.gotoliftpercentage"}
1916     ]
1917 }
1918

```

## 1919 8.23 Window Covering Cluster - Lift Percentage - Information

### 1920 8.23.1 Derived model

1921 The derived model: "zcl.windowcovering\_liftpercentage.info".

### 1922 8.23.2 Property definition

1923 Table 45 provides the detailed per Property mapping for "zcl.windowcovering\_liftpercentage.info".

1924 **Table 45 – The Property mapping for "zcl.windowcovering\_liftpercentage.info".**

Zigbee Property name	OCF Resource	To OCF	From OCF
CurrentPositionLiftPercentage	oic.r.openlevel	ocf.openLevel= CurrentPositionLiftPercentage	N/A

1925 Table 46 provides the details of the Properties that are part of  
1926 "zcl.windowcovering\_liftpercentage.info".

1927 **Table 46 – The Properties of "zcl.windowcovering\_liftpercentage.info".**

Zigbee Property name	Type	Required	Description
CurrentPositionLiftPercentage	integer	yes	Position as a percentage between InstalledOpenLimit-Lift and InstalledClosedLimit-Lift

### 1928 8.23.3 Derived model definition

```

1929 {
1930     "id":
1931     "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_liftpercentage.info.json#",
1932     "$schema": "http://json-schema.org/draft-04/schema#",
1933     "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1934     "title": "Window Covering Cluster - Lift Percentage - Information",
1935     "definitions": {
1936         "zcl.windowcovering_liftpercentage.info": {
1937             "type": "object",

```



```

1938     "properties": {
1939       "CurrentPositionLiftPercentage": {
1940         "type": "integer",
1941         "description": "Position as a percentage between InstalledOpenLimit-Lift and
1942 InstalledClosedLimit-Lift",
1943         "x-ocf-conversion": {
1944           "x-ocf-alias": "oic.r.openlevel",
1945           "x-to-ocf": [
1946             "ocf.openLevel= CurrentPositionLiftPercentage"
1947           ],
1948           "x-from-ocf": [
1949             "N/A"
1950           ]
1951         }
1952       }
1953     }
1954   },
1955 },
1956 "type": "object",
1957 "allOf": [
1958   {"$ref": "#/definitions/zcl.windowcovering_liftpercentage.info"}
1959 ],
1960 "required": ["CurrentPositionLiftPercentage"]
1961 }
1962

```

1963 **8.24 Window Covering Cluster - Lift Position - Control**

1964 **8.24.1 Derived model**

1965 The derived model: "zcl.windowcovering\_liftposition.control.gotoliftvalue".

1966 **8.24.2 Property definition**

1967 Table 47 provides the detailed per Property mapping for  
 1968 "zcl.windowcovering\_liftposition.control.gotoliftvalue".

1969 **Table 47 – The Property mapping for**  
 1970 **"zcl.windowcovering\_liftposition.control.gotoliftvalue".**

Zigbee Property name	OCF Resource	To OCF	From OCF
liftvalue	oic.r.openlevel	N/A	if ocf.openLevel is updated, lift value= ocf.openLevel.zcl.command.windowcovering::gotoliftvalue(lift value)

1971 Table 48 provides the details of the Properties that are part of  
 1972 "zcl.windowcovering\_liftposition.control.gotoliftvalue".

1973 **Table 48 – The Properties of "zcl.windowcovering\_liftposition.control.gotoliftvalue".**

Zigbee name	Property	Type	Required	Description
liftvalue		integer	no	Adjust the window at the lift value.

1974 **8.24.3 Derived model definition**

```

1975 {
1976   "id":
1977   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_liftposition.control.json#",
1978   "$schema": "http://json-schema.org/draft-04/schema#",
1979   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1980   "title": "Window Covering Cluster - Lift Position - Control",
1981   "definitions": {
1982     "zcl.windowcovering_liftposition.control.gotoliftvalue": {

```

```

1983     "properties": {
1984         "liftvalue": {
1985             "type": "integer",
1986             "description": "Adjust the window at the lift value.",
1987             "x-ocf-conversion": {
1988                 "x-ocf-alias": "oic.r.openlevel",
1989                 "x-from-ocf": [
1990                     "if ocf.openLevel is updated, lift value= ocf.openLevel.",
1991                     "zcl.command.windowcovering::gotoliftvalue(liftvalue)"
1992                 ],
1993                 "x-to-ocf": [
1994                     "N/A"
1995                 ]
1996             }
1997         }
1998     }
1999 },
2000 },
2001 "type": "object",
2002 "allOf": [
2003     {"$ref": "#/definitions/zcl.windowcovering_liftposition.control.gotoliftvalue"}
2004 ]
2005 }
2006

```

## 2007 8.25 Window Covering Cluster - Lift Position - Information

### 2008 8.25.1 Derived model

2009 The derived model: "zcl.windowcovering\_liftposition.info".

### 2010 8.25.2 Property definition

2011 Table 49 provides the detailed per Property mapping for "zcl.windowcovering\_liftposition.info".

2012 **Table 49 – The Property mapping for "zcl.windowcovering\_liftposition.info".**

Zigbee name	Property	OCF Resource	To OCF	From OCF
CurrentPosition-Lift		oic.r.openlevel	ocf.openLevel= CurrentPosition-Lift	N/A
InstalledClosedLimit-Lift		oic.r.openlevel	ocf.range[0]= InstalledClosedLimit-Lift	N/A
InstalledOpenLimit-Lift		oic.r.openlevel	ocf.range[1]= InstalledOpenLimit-Lift	N/A

2013 Table 50 provides the details of the Properties that are part of  
2014 "zcl.windowcovering\_liftposition.info".

2015 **Table 50 – The Properties of "zcl.windowcovering\_liftposition.info".**

Zigbee name	Property	Type	Required	Description
CurrentPosition-Lift		integer	yes	Position of Window Covering from the top of the shade (cm)
InstalledClosedLimit-Lift		integer	yes	Close limit for lifting the Window Covering (cm)
InstalledOpenLimit-Lift		integer	yes	Open limit for lifting the Window Covering (cm)

### 2016 8.25.3 Derived model definition

```
2017 {
2018   "id":
2019   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_liftposition.info.json#",
2020   "$schema": "http://json-schema.org/draft-04/schema#",
2021   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2022   "title": "Window Covering Cluster - Lift Position - Information",
2023   "definitions": {
2024     "zcl.windowcovering_liftposition.info": {
2025       "type": "object",
2026       "properties": {
2027         "InstalledClosedLimit-Lift": {
2028           "type": "integer",
2029           "description": "Close limit for lifting the Window Covering (cm)",
2030           "x-ocf-conversion": {
2031             "x-ocf-alias": "oic.r.openlevel",
2032             "x-to-ocf": [
2033               "ocf.range[0]= InstalledClosedLimit-Lift"
2034             ],
2035             "x-from-ocf": [
2036               "N/A"
2037             ]
2038           }
2039         },
2040         "InstalledOpenLimit-Lift": {
2041           "type": "integer",
2042           "description": "Open limit for lifting the Window Covering (cm)",
2043           "x-ocf-conversion": {
2044             "x-ocf-alias": "oic.r.openlevel",
2045             "x-to-ocf": [
2046               "ocf.range[1]= InstalledOpenLimit-Lift"
2047             ],
2048             "x-from-ocf": [
2049               "N/A"
2050             ]
2051           }
2052         },
2053         "CurrentPosition-Lift": {
2054           "type": "integer",
2055           "description": "Position of Window Covering from the top of the shade (cm)",
2056           "x-ocf-conversion": {
2057             "x-ocf-alias": "oic.r.openlevel",
2058             "x-to-ocf": [
2059               "ocf.openLevel= CurrentPosition-Lift"
2060             ],
2061             "x-from-ocf": [
2062               "N/A"
2063             ]
2064           }
2065         }
2066       }
2067     }
2068   },
2069   "type": "object",
2070   "allOf": [
2071     {"$ref": "#/definitions/zcl.windowcovering_liftposition.info"}
2072   ],
2073   "required": [ "InstalledClosedLimit-Lift", "InstalledOpenLimit-Lift", "CurrentPosition-Lift" ]
2074 }
2075
```

### 2076 8.26 Window Covering Cluster - Tilt Percentage - Control

#### 2077 8.26.1 Derived model

2078 The derived model: "zcl.windowcovering\_tiltpercentage.control.gotiltpercentage".

2079 **8.26.2 Property definition**

2080 Table 51 provides the detailed per Property mapping for  
 2081 "zcl.windowcovering\_tiltpercentage.control.gototiltpercentage".

2082 **Table 51 – The Property mapping for**  
 2083 **"zcl.windowcovering\_tiltpercentage.control.gototiltpercentage".**

Zigbee Property name	OCF Resource	To OCF	From OCF
percentagetiltvalue	oic.r.openlevel	N/A	if ocf.openLevel is updated, percentage tilt value = ocf.openLevel.zcl.command.windowcovering::gototiltpercentage(percentagetiltvalue)

2084 Table 52 provides the details of the Properties that are part of  
 2085 "zcl.windowcovering\_tiltpercentage.control.gototiltpercentage".

2086 **Table 52 – The Properties of**  
 2087 **"zcl.windowcovering\_tiltpercentage.control.gototiltpercentage".**

Zigbee name	Property	Type	Required	Description
	percentagetiltvalue	integer	no	Adjust the window at the percentage tilt value.

2088 **8.26.3 Derived model definition**

```

2089 {
2090   "id":
2091   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_tiltpercentage.control.json#"
2092 ,
2093   "$schema": "http://json-schema.org/draft-04/schema#",
2094   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2095   "title": "Window Covering Cluster - Tilt Percentage - Control",
2096   "definitions": {
2097     "zcl.windowcovering_tiltpercentage.control.gototiltpercentage": {
2098       "properties": {
2099         "percentagetiltvalue": {
2100           "type": "integer",
2101           "description": "Adjust the window at the percentage tilt value.",
2102           "x-ocf-conversion": {
2103             "x-ocf-alias": "oic.r.openlevel",
2104             "x-from-ocf": [
2105               "if ocf.openLevel is updated, percentage tilt value = ocf.openLevel.",
2106               "zcl.command.windowcovering::gototiltpercentage(percentagetiltvalue)"
2107             ],
2108             "x-to-ocf": [
2109               "N/A"
2110             ]
2111           }
2112         }
2113       }
2114     }
2115   },
2116   "type": "object",
2117   "allOf": [
2118     {"$ref": "#/definitions/zcl.windowcovering_tiltpercentage.control.gototiltpercentage"}
2119   ]
2120 }
2121 
```

2122 **8.27 Window Covering Cluster - Tilt Percentage - Information**

2123 **8.27.1 Derived model**

2124 The derived model: "zcl.windowcovering\_tiltpercentage.info".

2125 **8.27.2 Property definition**

2126 Table 53 provides the detailed per Property mapping for "zcl.windowcovering\_tiltpercentage.info".

2127 **Table 53 – The Property mapping for "zcl.windowcovering\_tiltpercentage.info".**

Zigbee Property name	OCF Resource	To OCF	From OCF
CurrentPositionTiltPercentage	oic.r.openlevel	ocf.openlevel=CurrentPositionTiltPercentage	N/A

2128 Table 54 provides the details of the Properties that are part of  
2129 "zcl.windowcovering\_tiltpercentage.info".

2130 **Table 54 – The Properties of "zcl.windowcovering\_tiltpercentage.info".**

Zigbee Property name	Type	Required	Description
CurrentPositionTiltPercentage	integer	yes	Tilt position as a percentage

2131 **8.27.3 Derived model definition**

```

2132 {
2133   "id":
2134   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_tiltpercentage.info.json#",
2135   "$schema": "http://json-schema.org/draft-04/schema#",
2136   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2137   "title": "Window Covering Cluster - Tilt Percentage - Information",
2138   "definitions": {
2139     "zcl.windowcovering_tiltpercentage.info": {
2140       "type": "object",
2141       "properties": {
2142         "CurrentPositionTiltPercentage": {
2143           "type": "integer",
2144           "description": "Tilt position as a percentage",
2145           "x-ocf-conversion": {
2146             "x-ocf-alias": "oic.r.openlevel",
2147             "x-to-ocf": [
2148               "ocf.openlevel=CurrentPositionTiltPercentage"
2149             ],
2150             "x-from-ocf": [
2151               "N/A"
2152             ]
2153           }
2154         }
2155       }
2156     }
2157   },
2158   "type": "object",
2159   "allOf": [
2160     {"$ref": "#/definitions/zcl.windowcovering_tiltpercentage.info"}
2161   ],
2162   "required": ["CurrentPositionTiltPercentage"]
2163 }
2164

```

2165 **8.28 Window Covering Cluster - Tilt Position - Control**

2166 **8.28.1 Derived model**

2167 The derived model: "zcl.windowcovering\_tiltposition.control.gototiltvalue".

2168 **8.28.2 Property definition**

2169 Table 55 provides the detailed per Property mapping for  
 2170 "zcl.windowcovering\_tiltposition.control.gototiltvalue".

2171 **Table 55 – The Property mapping for**  
 2172 **"zcl.windowcovering\_tiltposition.control.gototiltvalue".**

Zigbee Property name	OCF Resource	To OCF	From OCF
tiltvalue	oic.r.openlevel	N/A	if ocf.openLevel is updated, tiltvalue= ocf.openLevel.zb.command.windowcovering::gototiltvalue(tiltvalue)

2173 Table 56 provides the details of the Properties that are part of  
 2174 "zcl.windowcovering\_tiltposition.control.gototiltvalue".

2175 **Table 56 – The Properties of "zcl.windowcovering\_tiltposition.control.gototiltvalue".**

Zigbee Property name	Type	Required	Description
tiltvalue	integer	no	Adjust the window at the tilt value.

2176 **8.28.3 Derived model definition**

```

2177 {
2178   "id":
2179   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_tiltposition.control.json#",
2180   "$schema": "http://json-schema.org/draft-04/schema#",
2181   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2182   "title": "Window Covering Cluster - Tilt Position - Control",
2183   "definitions": {
2184     "zcl.windowcovering_tiltposition.control.gototiltvalue": {
2185       "properties": {
2186         "tiltvalue": {
2187           "type": "integer",
2188           "description": "Adjust the window at the tilt value.",
2189           "x-ocf-conversion": {
2190             "x-ocf-alias": "oic.r.openlevel",
2191             "x-from-ocf": [
2192               "if ocf.openLevel is updated, tiltvalue= ocf.openLevel.",
2193               "zb.command.windowcovering::gototiltvalue(tiltvalue)"
2194             ],
2195             "x-to-ocf": [
2196               "N/A"
2197             ]
2198           }
2199         }
2200       }
2201     }
2202   },
2203   "type": "object",
2204   "allOf": [
2205     {"$ref": "#/definitions/zcl.windowcovering_tiltposition.control.gototiltvalue"}
2206   ]
2207 }
2208
  
```

2209 **8.29 Window Covering Cluster - Tilt Position - Information**

2210 **8.29.1 Derived model**

2211 The derived model: "zcl.windowcovering\_tiltposition.info".

2212 **8.29.2 Property definition**

2213 Table 57 provides the detailed per Property mapping for "zcl.windowcovering\_tiltposition.info".

2214 **Table 57 – The Property mapping for "zcl.windowcovering\_tiltposition.info".**

Zigbee name	Property	OCF Resource	To OCF	From OCF
InstalledOpenLimit-Tilt		oic.r.openlevel	ocf.range[1]= InstalledOpenLimit-Tilt	N/A
CurrentPosition-Tilt		oic.r.openlevel	ocf.openlevel= CurrentPosition-Tilt	N/A

2215 Table 58 provides the details of the Properties that are part of  
2216 "zcl.windowcovering\_tiltposition.info".

2217 **Table 58 – The Properties of "zcl.windowcovering\_tiltposition.info".**

Zigbee name	Property	Type	Required	Description
InstalledOpenLimit-Tilt		integer	yes	Open limit for tilting the Window Covering (0.1 degree)
CurrentPosition-Tilt		integer	no	Tilt position of Window Covering from open (0.1 degree)

2218 **8.29.3 Derived model definition**

```

2219 {
2220   "id":
2221   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_tiltposition.info.json#",
2222   "$schema": "http://json-schema.org/draft-04/schema#",
2223   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2224   "title": "Window Covering Cluster - Tilt Position - Information",
2225   "definitions": {
2226     "zcl.windowcovering_tiltposition.info": {
2227       "type": "object",
2228       "properties": {
2229         "InstalledOpenLimit-Tilt": {
2230           "type": "integer",
2231           "description": "Close limit for tilting the Window Covering (0.1 degree)",
2232           "x-ocf-conversion": {
2233             "x-ocf-alias": "oic.r.openlevel",
2234             "x-to-ocf": [
2235               "ocf.range[0] = InstalledClosedLimit-Tilt"
2236             ],
2237             "x-from-ocf": [
2238               "N/A"
2239             ]
2240           }
2241         },
2242         "InstalledOpenLimit-Tilt": {
2243           "type": "integer",
2244           "description": "Open limit for tilting the Window Covering (0.1 degree)",
2245           "x-ocf-conversion": {
2246             "x-ocf-alias": "oic.r.openlevel",
2247             "x-to-ocf": [
2248               "ocf.range[1]= InstalledOpenLimit-Tilt"
2249             ],
2250             "x-from-ocf": [
2251               "N/A"
2252             ]
2253           }
2254         }
2255       }
2256     }
2257   }

```

```

2254     },
2255     "CurrentPosition-Tilt ": {
2256       "type": "integer",
2257       "description": "Tilt position of Window Covering from open (0.1 degree)",
2258       "x-ocf-conversion": {
2259         "x-ocf-alias": "oic.r.openlevel",
2260         "x-to-ocf": [
2261           "ocf.openlevel= CurrentPosition-Tilt"
2262         ],
2263         "x-from-ocf": [
2264           "N/A"
2265         ]
2266       }
2267     }
2268   }
2269 },
2270 },
2271 "type": "object",
2272 "allOf": [
2273   {"$ref": "#/definitions/zcl.windowcovering_tiltposition.info"}
2274 ],
2275 "required": [ "InstalledClosedLimit-Tilt", "InstalledOpenLimit-Tilt", "CurrentPosition-Tilt" ]
2276 }
2277
2278

```