

OCF Resource to Zigbee Cluster Mapping Specification

VERSION 2.0.3 | June 2019



OPEN CONNECTIVITY
FOUNDATION™

CONTACT 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

21

22 CONTENTS

23

24 1 Scope 1

25 2 Normative references 1

26 3 Terms, definitions symbols and abbreviations 1

27 4 Document conventions and organization 2

28 4.1 Conventions 2

29 4.2 Notation..... 2

30 5 Theory of Operation 3

31 5.1 Interworking Approach..... 3

32 5.2 Mapping Syntax..... 3

33 5.2.1 Introduction 3

34 5.2.2 General..... 3

35 5.2.3 Value Assignment 3

36 5.2.4 Property Naming 3

37 5.2.5 Range 3

38 5.2.6 Arrays 3

39 5.2.7 Default Mapping 3

40 5.2.8 Conditional Mapping..... 3

41 5.2.9 Method Invocation..... 4

42 6 Device Type Mapping..... 4

43 6.1 Introduction 4

44 6.2 Zigbee Device Types to OCF Device Types..... 4

45 7 Resource to ZigBee Cluster Equivalence..... 5

46 7.1 Introduction 5

47 7.2 Zigbee Clusters to OCF Resources..... 5

48 7.2.1 Introduction 5

49 7.2.2 On/off..... 6

50 7.2.3 Level Control..... 6

51 7.2.4 Color Control..... 6

52 7.2.5 Thermostat..... 7

53 7.2.6 Window Covering 7

54 7.2.7 Temperature Measurement 8

55 7.2.8 Occupancy Sensing..... 8

56 7.2.9 IAS Zone..... 8

57 8 Detailed Mapping APIs..... 9

58 8.1.1 Introduction 9

59 8.2 Color Control Cluster - Color Space - Control 9

60 8.2.1 Derived model..... 9

61 8.2.2 Property definition 9

62 8.2.3 Derived model definition..... 10

63	8.3	Color Control Cluster - Color Space - Information	10
64	8.3.1	Derived model	10
65	8.3.2	Property definition	10
66	8.3.3	Derived model definition	11
67	8.4	Color Control Cluster - Color Temperature - Information	12
68	8.4.1	Derived model	12
69	8.4.2	Property definition	12
70	8.4.3	Derived model definition	12
71	8.5	Color Control Cluster - Color Temperature - Information	13
72	8.5.1	Derived model	13
73	8.5.2	Property definition	13
74	8.5.3	Derived model definition	13
75	8.6	Color Control Cluster - Hue and Saturation - Control	14
76	8.6.1	Derived model	14
77	8.6.2	Property definition	14
78	8.6.3	Derived model definition	15
79	8.7	Color Control Cluster - Hue and Saturation - Information	16
80	8.7.1	Derived model	16
81	8.7.2	Property definition	16
82	8.7.3	Derived model definition	16
83	8.8	IAS Zone Cluster - Control.....	17
84	8.8.1	Derived model	17
85	8.8.2	Property definition	17
86	8.8.3	Derived model definition	17
87	8.9	IAS Zone Cluster - Information	18
88	8.9.1	Derived model	18
89	8.9.2	Property definition	18
90	8.9.3	Derived model definition	24
91	8.10	Level Control Cluster - Control	27
92	8.10.1	Derived model	27
93	8.10.2	Property definition	27
94	8.10.3	Derived model definition	28
95	8.11	Level Control Cluster - Information	28
96	8.11.1	Derived model	28
97	8.11.2	Property definition	28
98	8.11.3	Derived model definition	29
99	8.12	Occupancy Sensing Cluster - Information	29
100	8.12.1	Derived model	29
101	8.12.2	Property definition	29
102	8.12.3	Derived model definition	30
103	8.13	On/Off Cluster - Control.....	30
104	8.13.1	Derived model	30
105	8.13.2	Property definition	30
106	8.13.3	Derived model definition	31

107	8.14	On/off Cluster - Information	32
108	8.14.1	Derived model	32
109	8.14.2	Property definition	32
110	8.14.3	Derived model definition	32
111	8.15	Temperature Measurement Cluster - Information	33
112	8.15.1	Derived model	33
113	8.15.2	Property definition	33
114	8.15.3	Derived model definition	33
115	8.16	Thermostat Cluster - Cool - Control	34
116	8.16.1	Derived model	34
117	8.16.2	Property definition	34
118	8.16.3	Derived model definition	35
119	8.17	Thermostat Cluster - Current Temperature - Information	35
120	8.17.1	Derived model	35
121	8.17.2	Property definition	36
122	8.17.3	Derived model definition	36
123	8.18	Thermostat Cluster - Heat - Control	36
124	8.18.1	Derived model	36
125	8.18.2	Property definition	36
126	8.18.3	Derived model definition	37
127	8.19	Window Covering Cluster - Configuration - Control	37
128	8.19.1	Derived model	37
129	8.19.2	Property definition	38
130	8.19.3	Derived model definition	39
131	8.20	Window Covering Cluster - Configuration - Information	40
132	8.20.1	Derived model	40
133	8.20.2	Property definition	40
134	8.20.3	Derived model definition	44
135	8.21	Window Covering Cluster - Lift Percentage - Control	46
136	8.21.1	Derived model	46
137	8.21.2	Property definition	46
138	8.21.3	Derived model definition	46
139	8.22	Window Covering Cluster - Lift Percentage - Information	47
140	8.22.1	Derived model	47
141	8.22.2	Property definition	47
142	8.22.3	Derived model definition	47
143	8.23	Window Covering Cluster - Lift Position - Control	48
144	8.23.1	Derived model	48
145	8.23.2	Property definition	48
146	8.23.3	Derived model definition	48
147	8.24	Window Covering Cluster - Lift Position - Information	49
148	8.24.1	Derived model	49
149	8.24.2	Property definition	49
150	8.24.3	Derived model definition	50

151	8.25	Window Covering Cluster - Tilt Percentage - Control	50
152	8.25.1	Derived model	50
153	8.25.2	Property definition	51
154	8.25.3	Derived model definition	51
155	8.26	Window Covering Cluster - Tilt Percentage - Information	52
156	8.26.1	Derived model	52
157	8.26.2	Property definition	52
158	8.26.3	Derived model definition	52
159	8.27	Window Covering Cluster - Tilt Position - Control	52
160	8.27.1	Derived model	52
161	8.27.2	Property definition	53
162	8.27.3	Derived model definition	53
163	8.28	Window Covering Cluster - Tilt Position - Information	53
164	8.28.1	Derived model	53
165	8.28.2	Property definition	54
166	8.28.3	Derived model definition	54
167			
168			

169 Figures

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

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

210	Table 38 – The Properties of "zcl.thermostat_heat.control.setpointraiseLower".	37
211	Table 39 – The Property mapping for "zcl.windowcovering_conf.control".	38
212	Table 40 – The Properties of "zcl.windowcovering_conf.control".	38
213	Table 41 – The Property mapping for "zcl.windowcovering_conf.info".	40
214	Table 42 – The Properties of "zcl.windowcovering_conf.info".	42
215	Table 43 – The Property mapping for	
216	"/zcl.windowcovering_liftpercentage.control.gotoliftpercentage".	46
217	Table 44 – The Properties of	
218	"/zcl.windowcovering_liftpercentage.control.gotoliftpercentage".	46
219	Table 45 – The Property mapping for "zcl.windowcovering_liftpercentage.info".	47
220	Table 46 – The Properties of "zcl.windowcovering_liftpercentage.info".	47
221	Table 47 – The Property mapping for	
222	"zcl.windowcovering_liftposition.control.gotoliftvalue".	48
223	Table 48 – The Properties of "zcl.windowcovering_liftposition.control.gotoliftvalue".	48
224	Table 49 – The Property mapping for "/zcl.windowcovering_liftposition.info".	49
225	Table 50 – The Properties of "/zcl.windowcovering_liftposition.info".	49
226	Table 51 – The Property mapping for	
227	"zcl.windowcovering_tiltpercentage.control.gototiltpercentage".	51
228	Table 52 – The Properties of	
229	"zcl.windowcovering_tiltpercentage.control.gototiltpercentage".	51
230	Table 53 – The Property mapping for "zcl.windowcovering_tiltpercentage.info".	52
231	Table 54 – The Properties of "zcl.windowcovering_tiltpercentage.info".	52
232	Table 55 – The Property mapping for	
233	"zcl.windowcovering_tiltposition.control.gototiltvalue".	53
234	Table 56 – The Properties of "zcl.windowcovering_tiltposition.control.gototiltvalue".	53
235	Table 57 – The Property mapping for "zcl.windowcovering_tiltposition.info".	54
236	Table 58 – The Properties of "zcl.windowcovering_tiltposition.info".	54
237		
238		

239 **1 Scope**

240 This document provides detailed mapping information between Zigbee defined Clusters and OCF
241 defined Resources.

242 **2 Normative references**

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

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

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

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

259 ISO/IEC 30118-4:2019, Information technology – Open Connectivity Foundation (OCF)
260 Specification – Part 4: Resource type specification
261 <https://www.iso.org/standard/74241.html>
262 Latest version available at:
263 https://openconnectivity.org/specs/OCF_Resource_Type_Specification.pdf

264 ISO/IEC 30118-5:2019, Information technology – Open Connectivity Foundation (OCF)
265 Specification – Part 5: Smart home device specification
266 <https://www.iso.org/standard/74242.html>
267 Latest version available at: https://openconnectivity.org/specs/OCF_Device_Specification.pdf

268 Derived Models for Interoperability between IoT Ecosystems, Stevens & Merriam, March 2016
269 [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)
270 [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)

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

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

275 **3 Terms, definitions symbols and abbreviations**

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

278 ISO and IEC maintain terminological databases for use in standardization at the following
279 addresses:

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

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

282 **4 Document conventions and organization**

283 **4.1 Conventions**

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

288 **4.2 Notation**

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

291 Required (or shall or mandatory).

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

295 Recommended (or should).

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

303 Allowed (or allowed).

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

307 Conditionally allowed (CA)

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

310 Conditionally required (CR)

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

314 DEPRECATED

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

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

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

323 **5 Theory of Operation**

324 **5.1 Interworking Approach**

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

327 **5.2 Mapping Syntax**

328 **5.2.1 Introduction**

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

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

336 **5.2.2 General**

337 All statements are terminated with a carriage return.

338 **5.2.3 Value Assignment**

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

341 **5.2.4 Property Naming**

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

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

347 **5.2.5 Range**

348 The range on the OCF side is fixed.

349 **5.2.6 Arrays**

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

352 **5.2.7 Default Mapping**

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

356 **5.2.8 Conditional Mapping**

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

358 If "condition", then "mapping".

359 is applied.

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

361 **5.2.9 Method Invocation**

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

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

368 **6 Device Type Mapping**

369 **6.1 Introduction**

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

371 **6.2 Zigbee Device Types to OCF Device Types**

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

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

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

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

376 **7 Resource to ZigBee Cluster Equivalence**

377 **7.1 Introduction**

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

381 **7.2 Zigbee Clusters to OCF Resources**

382 **7.2.1 Introduction**

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

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

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

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

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

388

389 **7.2.2 On/off**

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

397 **7.2.3 Level Control**

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

405 **7.2.4 Color Control**

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

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

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

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

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

433 **7.2.5 Thermostat**

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

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

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

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

451 **7.2.6 Window Covering**

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

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

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

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

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

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

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

494 **7.2.7 Temperature Measurement**

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

501 **7.2.8 Occupancy Sensing**

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

508 **7.2.9 IAS Zone**

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

515 **8 Detailed Mapping APIs**

516 **8.1 below**

517 **8.2 Introduction**

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

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

523 **8.3 Color Control Cluster - Color Space - Control**

524 **8.3.1 Derived model**

525 The derived model: "zcl.colorcontrol_csc.control.movetocolor".

526 **8.3.2 Property definition**

527 Table 3 provides the detailed per Property mapping for "zcl.colorcontrol_csc.control.movetocolor".

528 **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= transitiontime=0zcl.command.colorcontrol::movetocolor(colorx,colory,transitiontime). & ocf.csc[1]*65536
colorx	oic.r.colour.csc	N/A	colorx transitiontime=0zcl.command.colorcontrol::movetocolor(colorx,colory,transitiontime). =ocf.csc[0]*65536 &

529 Table 4 provides the details of the Properties that are part of
 530 "zcl.colorcontrol_csc.control.movetocolor".

531 **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.

532 **8.3.3 Derived model definition**

```

533 {
534   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_csc.control.json#",
535   "$schema": "http://json-schema.org/draft-04/schema#",
536   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
537   "title": "Color Control Cluster - Color Space - Control",
538   "definitions": {
539     "zcl.colorcontrol_csc.control.movetocolor": {
540       "properties": {
541         "colorx": {
542           "type": "number",
543           "description": "Move to certain value(s) of color coordinates as fast as possible with
544 transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
545           "x-ocf-conversion": {
546             "x-ocf-alias": "oic.r.colour.csc",
547             "x-from-ocf": [
548               "colorx =ocf.csc[0]*65536 & transitiontime=0",
549               "zcl.command.colorcontrol::movetocolor(colorx,colory,transitiontime)."
550             ],
551             "x-to-ocf": [
552               "N/A"
553             ]
554           }
555         },
556         "colory": {
557           "type": "number",
558           "description": "Move to certain value(s) of color coordinates as fast as possible with
559 transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
560           "x-ocf-conversion": {
561             "x-ocf-alias": "oic.r.colour.csc",
562             "x-from-ocf": [
563               "colory= ocf.csc[1]*65536 & transitiontime=0",
564               "zcl.command.colorcontrol::movetocolor(colorx,colory,transitiontime)."
565             ],
566             "x-to-ocf": [
567               "N/A"
568             ]
569           }
570         }
571       }
572     },
573     "type": "object",
574     "allOf": [
575       {"$ref": "#/definitions/zcl.colorcontrol_csc.control.movetocolor"}
576     ]
577   }
578 }
579

```

580 **8.4 Color Control Cluster - Color Space - Information**

581 **8.4.1 Derived model**

582 The derived model: "zcl.colorcontrol_csc.info".

583 **8.4.2 Property definition**

584 Table 5 provides the detailed per Property mapping for "zcl.colorcontrol_csc.info".

585 **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

586 Table 6 provides the details of the Properties that are part of "zcl.colorcontrol_csc.info".

587 **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

588 **8.4.3 Derived model definition**

```

589 {
590   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_csc.info.json#",
591   "$schema": "http://json-schema.org/draft-04/schema#",
592   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
593   "title": "Color Control Cluster - Color Space - Information",
594   "definitions": {
595     "zcl.colorcontrol_csc.info": {
596       "type": "object",
597       "properties": {
598         "currentX": {
599           "type": "integer",
600           "description": "current value of the normalized chromaticity value x, as defined in the
601 CIE xy Color Space",
602           "x-ocf-conversion": {
603             "x-ocf-alias": "oic.r.colour.csc",
604             "x-to-ocf": [
605               "ocf.csc[0] = currentX/65536"
606             ],
607             "x-from-ocf": [
608               "N/A"
609             ]
610           }
611         },
612         "currentY": {
613           "type": "integer",
614           "description": "current value of the normalized chromaticity value y, as defined in the
615 CIE xy Color Space",
616           "x-ocf-conversion": {
617             "x-ocf-alias": "oic.r.colour.csc",
618             "x-to-ocf": [
619               "ocf.csc[1] = currentY/65536"
620             ],
621             "x-from-ocf": [
622               "N/A"
623             ]
624           }
625         }
626       }
627     }
628   },
629   "type": "object",
630   "allOf": [
631     {"$ref": "#/definitions/zcl.colorcontrol_csc.info"}
632   ],
633   "required": ["currentx", "currenty"]
634 }
635

```

636 **8.5 Color Control Cluster - Color Temperature - Information**

637 **8.5.1 Derived model**

638 The derived model: "zcl.colorcontrol_ct.control.movetocolortemperature".

639 **8.5.2 Property definition**

640 Table 7 provides the detailed per Property mapping for
641 "zcl.colorcontrol_ct.control.movetocolortemperature".

642 **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)

643 Table 8 provides the details of the Properties that are part of
644 "zcl.colorcontrol_ct.control.movetocolortemperature".

645 **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.

646 **8.5.3 Derived model definition**

```

647 {
648   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_ct.control.json#",
649   "$schema": "http://json-schema.org/draft-04/schema#",
650   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
651   "title": "Color Control Cluster - Color Temperature - Information",
652   "definitions": {
653     "zcl.colorcontrol_ct.control.movetocolortemperature": {
654       "properties": {
655         "colortemperature": {
656           "type": "integer",
657           "description": "Move to certain value of colortemperature as fast as possible with
658 transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
659           "x-ocf-conversion": {
660             "x-ocf-alias": "oic.r.colour.colourtemperature",
661             "x-from-ocf": [
662               "colourtemperature=ocf.ct & transitiontime=0",
663               "zcl.command.colorcontrol::movetocolortemperature(colortemperature,transitiontime)"
664             ],
665             "x-to-ocf": [
666               "N/A"
667             ]
668           }
669         }
670       }
671     },
672     "type": "object",
673     "allOf": [
674

```

```

675     {"$ref": "#/definitions/zcl.colorcontrol_ct.control.movetocolortemperature"}
676   ]
677 }
678

```

679 8.6 Color Control Cluster - Color Temperature - Information

680 8.6.1 Derived model

681 The derived model: "zcl.colorcontrol_ct.info".

682 8.6.2 Property definition

683 Table 9 provides the detailed per Property mapping for "zcl.colorcontrol_ct.info".

684 **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

685 Table 10 provides the details of the Properties that are part of "zcl.colorcontrol_ct.info".

686 **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

687 8.6.3 Derived model definition

```

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

```

```

710     "colortempphysicalmin": {
711       "type": "integer",
712       "description": "minimum mired value supported by the hardware",
713       "x-ocf-conversion": {
714         "x-ocf-alias": "oic.r.colour.colourtemperature",
715         "x-to-ocf": [
716           "ocf.range[0] = colortempphysicalmin"
717         ],
718         "x-from-ocf": [
719           "N/A"
720         ]
721       }
722     },
723     "colorphysicalmax": {
724       "type": "integer",
725       "description": "maximum mired value supported by the hardware",
726       "x-ocf-conversion": {
727         "x-ocf-alias": "oic.r.colour.colourtemperature",
728         "x-to-ocf": [
729           "ocf.range[1] = colorphysicalmax"
730         ],
731         "x-from-ocf": [
732           "N/A"
733         ]
734       }
735     }
736   }
737 }
738 },
739 "type": "object",
740 "allof": [
741   {"$ref": "#/definitions/zcl.colorcontrol_ct.info"}
742 ],
743 "required": ["colortemperaturemired", "colortempphysicalmin", "colortempphysicalmax" ]
744 }
745

```

746 8.7 Color Control Cluster - Hue and Saturation - Control

747 8.7.1 Derived model

748 The derived model: "zcl.colorcontrol_hs.control.movetohueandsaturation".

749 8.7.2 Property definition

750 Table 11 provides the detailed per Property mapping for
751 "zcl.colorcontrol_hs.control.movetohueandsaturation".

752 **Table 11 – The Property mapping for**
753 **"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)

754 Table 12 provides the details of the Properties that are part of
755 "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.

8.7.3 Derived model definition

```

757 {
758   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_hs.control.json#",
759   "$schema": "http://json-schema.org/draft-04/schema#",
760   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
761   "title": "Color Control Cluster - Hue and Saturation - Control",
762   "definitions": {
763     "zcl.colorcontrol_hs.control.movetohueandsaturation": {
764       "properties": {
765         "hue": {
766           "type": "integer",
767           "description": "Move to certain value(s) of hue or saturation or both as fast as possible
768 with transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
769           "x-ocf-conversion": {
770             "x-ocf-alias": "oic.r.colour.hs",
771             "x-from-ocf": [
772               "hue=ocf.hue/360 * 254 & transitiontime=0",
773               "zcl.command.colorcontrol::movetohueandsaturation(hue,saturation,transitiontime)"
774             ],
775             "x-to-ocf": [
776               "N/A"
777             ]
778           }
779         },
780         "saturation": {
781           "type": "integer",
782           "description": "Move to certain value(s) of hue or saturation or both as fast as possible
783 with transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
784           "x-ocf-conversion": {
785             "x-ocf-alias": "oic.r.colour.hs",
786             "x-from-ocf": [
787               "saturation=ocf.saturation & transitiontime=0",
788               "zcl.command.colorcontrol::movetohueandsaturation(hue,saturation,transitiontime)"
789             ],
790             "x-to-ocf": [
791               "N/A"
792             ]
793           }
794         }
795       }
796     }
797   },
798   "type": "object",
799   "allOf": [
800     {"$ref": "#/definitions/zcl.colorcontrol_hs.control.movetohueandsaturation"}
801   ]
802 }

```


803 }
804

805 8.8 Color Control Cluster - Hue and Saturation - Information

806 8.8.1 Derived model

807 The derived model: "zcl.colorcontrol_hs.info".

808 8.8.2 Property definition

809 Table 13 provides the detailed per Property mapping for "zcl.colorcontrol_hs.info".

810 **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

811 Table 14 provides the details of the Properties that are part of "zcl.colorcontrol_hs.info".

812 **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

813 8.8.3 Derived model definition

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

```

```

845         "N/A"
846     ]
847 }
848 }
849 }
850 }
851 },
852 "type": "object",
853 "allOf": [
854   {"$ref": "#/definitions/zcl.colorcontrol_hs.info"}
855 ],
856 "required": ["currenthue", "currentsaturation"]
857 }
858

```

859 8.9 IAS Zone Cluster - Control

860 8.9.1 Derived model

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

862 8.9.2 Property definition

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

864 **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)

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

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

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

867 8.9.3 Derived model definition

```

868 {
869   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.iaszone.control.json#",
870   "$schema": "http://json-schema.org/draft-04/schema#",
871   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
872   "title": "IAS Zone Cluster - Control",
873   "definitions": {
874     "zcl.iaszone.control": {
875       "properties": {
876         "currentzonesensitivitylevel": {
877           "type": "integer",
878           "description": "Set a sensitivity level of IAS Zone",
879           "x-ocf-conversion": {
880             "x-ocf-alias": "oic.r.ias.zone",
881             "x-from-ocf": [
882               "currentzonesensitivitylevel = ocf.currentzonesensitivitylevel",
883               "zcl.command.general::write(currentzonesensitivitylevel)"
884             ],
885             "x-to-ocf": [
886               "N/A"
887             ]
888           }
889         }
890       }
891     }
892   }

```

```

892     },
893     "type": "object",
894     "allOf": [
895         {"$ref": "#/definitions/zcl.iaszone.control"}
896     ]
897 }
898

```

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.numzonesensitivitylevels_supported=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 & zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=["]if zonetype=0x000d & zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=['intrusion']if zonetype=0x000d & zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0x000d & zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=['presence']if zonetype=0x000d & zonestatus=xxxxxxxxxxx11, ocf.zonestatus.alarms=['intrusion','presence']if zonetype=0x0015 & zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=["]if zonetype=0x0015 & zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=['1stportalopenclose']if zonetype=0x0015 & zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0x0015 & zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=['2ndportalopenclose']if zonetype=0x0015 & zonestatus=xxxxxxxxxxx11, ocf.zonestatus.alarms=['1stportalopenclose','2n dportalopenclose']if zonetype=0x0028 & zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=["]if zonetype=0x0028 & zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=['fire']if zonetype=0x0028 & zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0x0028 & zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=["]if zonetype=0x002a & zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=["]if zonetype=0x002a & zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=['wateroverflow']if zonetype=0x002a & zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0x002a & zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=["]if zonetype=0x002b & zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=["]if zonetype=0x002b & zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=['CO']if zonetype=0x002b & zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0x002b & zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=['cooking']if zonetype=0x002b & </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 & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['panic']if zonetype=0x021d & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0x021d & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['emergency']if zonetype=0x021d & zonestatus=xxxxxxxxxxxx11, ocf.zonestatus.alarms=['panic','emergency']if zonetype=0x0225 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=["]if zonetype=0x0225 & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['glassbreak']if zonetype=0x0225 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0x0225 & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=["]if zonetype=0x0226 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=["]if zonetype=0x0226 & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=["]if zonetype=0x0226 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0x0226 & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=["]if zonetype=0x0229 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=["]if zonetype=0x0229 & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=["]if zonetype=0x0229 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0x0229 & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=["]if zonetype=0xffff & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=["]if zonetype=0xffff & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=["]if zonetype=0xffff & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=["]if zonetype=0xffff & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=["]if zonestatus=xxxxxxxxxxxx0xx, ocf.zonestatus.tamper=falseif zonestatus=xxxxxxxxxxxx1xx, ocf.zonestatus.tamper=trueif zonestatus=xxxxxxxxxxxx0xxx, ocf.zonebattery.charge=100 & ocf.zonebattery.lowbattery=falseif zonestatus=xxxxxxxxxxxx1xxx, ocf.zonebattery.charge=100 & 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

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

906 **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

907 8.10.3 Derived model definition

```
908 {
909   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.iaszone.info.json#",
910   "$schema": "http://json-schema.org/draft-04/schema#",
911   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
912   "title": "IAS Zone Cluster - Information",
913   "definitions": {
914     "zcl.iaszone.info": {
915       "type": "object",
916       "properties": {
917         "zonestate": {
918           "type": "boolean",
919           "description": "Enrollment status of IAS Zone false=not enrolled, true=enrolled",
920           "x-ocf-conversion": {
921             "x-ocf-alias": "oic.r.iaszone",
922             "x-to-ocf": [
923               "if zonestate=0x00, ocf.zonestate=false",
924               "if zonestate=0x01, ocf.zonestate=true"
925             ],
926             "x-from-ocf": [
927               "N/A"
928             ]
929           }
930         },
931         "zonetype": {
932           "type": "string",
933           "description": "Zonetype and Meaning of Alarm1 and Alarm2 zonestatus",
934           "x-ocf-conversion": {
935             "x-ocf-alias": "oic.r.iaszone",
936             "x-to-ocf": [
937               "if zonetype=0x0000, ocf.zonetype=Standard CIE",
938               "if zonetype=0x000d, ocf.zonetype=Motion sensor",
939               "if zonetype=0x0015, ocf.zonetype=Contact switch",
940               "if zonetype=0x0028, ocf.zonetype=Fire sensor",
941               "if zonetype=0x002a, ocf.zonetype=Water sensor",
942               "if zonetype=0x002b, ocf.zonetype=Carbon Monoxide (CO) sensor",
943               "if zonetype=0x002c, ocf.zonetype=Personal emergency device",
944               "if zonetype=0x002d, ocf.zonetype=Vibration/Movement sensor",
945               "if zonetype=0x010f, ocf.zonetype=Remote Control",
946               "if zonetype=0x0115, ocf.zonetype=Key fob",
947               "if zonetype=0x021d, ocf.zonetype=Keypad",
948               "if zonetype=0x0225, ocf.zonetype=Standard Warning Device",
949               "if zonetype=0x0226, ocf.zonetype=Glass break sensor",
950               "if zonetype=0x0229, ocf.zonetype=Security repeater",
951               "if zonetype=0xffff, ocf.zonetype=Invalid Zone Type"
952             ],
953             "x-from-ocf": [
954               "N/A"
955             ]
956           }
957         },
958         "zonestatus": {
959           "type": "array",
960           "items": {
961             "type": "integer"
962           },
963           "description": "x is a variable. zonestatus in Zigbee maps to zonestatus, zonebattery,
964 and zonepowersource in OCF. Data type of zonestatus in Zigbee is 16 bitmap (xxxxxxxxxxxxxxxx) : bit
965 0 = Alarm1, bit 1 = Alarm2, bit 2 = Tamper, bit 3 = Battery, bit 4 = Supervision reports, bit 5 =
966 Restore reports, bit 6 = Trouble, bit 7 = AC (mains), bit 8 = Test, bit 9 = Battery Defect.
967 Alarm1 : 1 = opened or alarmed 0 = closed or not alarmed, Alarm2 : 1 = opened or alarmed 0 = closed
968 or not alarmed, Tamper : 1 = Tampered 0 = Not tampered, Battery : 1 = Low battery 0 = Battery OK,
969 Supervision reports : 1 = Reports 0 = Does not report, Restore reports : 1 = Reports restore 0 =
970 Does not report restore, Trouble : 1 = Trouble/Failure 0 = OK, AC (mains) : 1 = AC/Mains fault 0 =
971 AC/Mains OK, Test : 1 = Sensor is in test mode 0 = Sensor is in operation mode, Battery Defect : 1
972 = Sensor detects a defective battery 0 = Sensor battery is functioning.",
973           "x-ocf-conversion": {
974             "x-ocf-alias": "oic.r.iaszone",
975             "x-to-ocf": [
976
```

```

977         "if zonetype=0x0000 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
978         "if zonetype=0x0000 & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['system']",
979         "if zonetype=0x0000 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
980         "if zonetype=0x0000 & zonestatus=xxxxxxxxxxxxlx, ocf.zonestatus.alarms=[' ']",
981
982         "if zonetype=0x000d & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
983         "if zonetype=0x000d & zonestatus=xxxxxxxxxxxxx1,
984 ocf.zonestatus.alarms=['intrusion']",
985         "if zonetype=0x000d & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
986         "if zonetype=0x000d & zonestatus=xxxxxxxxxxxxlx,
987 ocf.zonestatus.alarms=['presence']",
988         "if zonetype=0x000d & zonestatus=xxxxxxxxxxxxl1,
989 ocf.zonestatus.alarms=['intrusion','presence']",
990
991         "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
992         "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxxx1,
993 ocf.zonestatus.alarms=['1stportalopenclose']",
994         "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
995         "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxxlx,
996 ocf.zonestatus.alarms=['2ndportalopenclose']",
997         "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxxl1,
998 ocf.zonestatus.alarms=['1stportalopenclose','2ndportalopenclose']",
999
1000        "if zonetype=0x0028 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1001        "if zonetype=0x0028 & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['fire']",
1002        "if zonetype=0x0028 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1003        "if zonetype=0x0028 & zonestatus=xxxxxxxxxxxxlx, ocf.zonestatus.alarms=[' ']",
1004
1005        "if zonetype=0x002a & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1006        "if zonetype=0x002a & zonestatus=xxxxxxxxxxxxx1,
1007 ocf.zonestatus.alarms=['wateroverflow']",
1008        "if zonetype=0x002a & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1009        "if zonetype=0x002a & zonestatus=xxxxxxxxxxxxlx, ocf.zonestatus.alarms=[' ']",
1010
1011        "if zonetype=0x002b & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1012        "if zonetype=0x002b & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['CO']",
1013        "if zonetype=0x002b & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1014        "if zonetype=0x002b & zonestatus=xxxxxxxxxxxxlx,
1015 ocf.zonestatus.alarms=['cooking']",
1016        "if zonetype=0x002b & zonestatus=xxxxxxxxxxxxl1,
1017 ocf.zonestatus.alarms=['CO','cooking']",
1018
1019        "if zonetype=0x002c & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1020        "if zonetype=0x002c & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['fall']",
1021        "if zonetype=0x002c & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1022        "if zonetype=0x002c & zonestatus=xxxxxxxxxxxxlx,
1023 ocf.zonestatus.alarms=['emergencybutton']",
1024        "if zonetype=0x002c & zonestatus=xxxxxxxxxxxxl1,
1025 ocf.zonestatus.alarms=['fall','emergencybutton']",
1026
1027        "if zonetype=0x002d & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1028        "if zonetype=0x002d & zonestatus=xxxxxxxxxxxxx1,
1029 ocf.zonestatus.alarms=['movement']",
1030        "if zonetype=0x002d & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1031        "if zonetype=0x002d & zonestatus=xxxxxxxxxxxxlx,
1032 ocf.zonestatus.alarms=['vibration']",
1033        "if zonetype=0x002d & zonestatus=xxxxxxxxxxxxl1,
1034 ocf.zonestatus.alarms=['movement','vibration']",
1035
1036        "if zonetype=0x010f & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1037        "if zonetype=0x010f & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['panic']",
1038        "if zonetype=0x010f & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1039        "if zonetype=0x010f & zonestatus=xxxxxxxxxxxxlx,
1040 ocf.zonestatus.alarms=['emergency']",
1041        "if zonetype=0x010f & zonestatus=xxxxxxxxxxxxl1,
1042 ocf.zonestatus.alarms=['panic','emergency']",
1043
1044        "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1045        "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['panic']",
1046        "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1047        "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxxlx,

```

```

1048 ocf.zonestatus.alarms=['emergency']",
1049     "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxx11,
1050 ocf.zonestatus.alarms=['panic','emergency']",
1051
1052     "if zonetype=0x021d & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
1053     "if zonetype=0x021d & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['panic']",
1054     "if zonetype=0x021d & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
1055     "if zonetype=0x021d & zonestatus=xxxxxxxxxxxx1x,
1056 ocf.zonestatus.alarms=['emergency']",
1057     "if zonetype=0x021d & zonestatus=xxxxxxxxxxxx11,
1058 ocf.zonestatus.alarms=['panic','emergency']",
1059
1060     "if zonetype=0x0225 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
1061     "if zonetype=0x0225 & zonestatus=xxxxxxxxxxxx1,
1062 ocf.zonestatus.alarms=['glassbreak']",
1063     "if zonetype=0x0225 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
1064     "if zonetype=0x0225 & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['']",
1065
1066     "if zonetype=0x0226 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
1067     "if zonetype=0x0226 & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['']",
1068     "if zonetype=0x0226 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
1069     "if zonetype=0x0226 & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['']",
1070
1071     "if zonetype=0x0229 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
1072     "if zonetype=0x0229 & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['']",
1073     "if zonetype=0x0229 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
1074     "if zonetype=0x0229 & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['']",
1075
1076     "if zonetype=0xffff & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
1077     "if zonetype=0xffff & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['']",
1078     "if zonetype=0xffff & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
1079     "if zonetype=0xffff & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['']",
1080
1081     "if zonestatus=xxxxxxxxxx0xx, ocf.zonestatus.tamper=false",
1082     "if zonestatus=xxxxxxxxxx1xx, ocf.zonestatus.tamper=true",
1083
1084     "if zonestatus=xxxxxxxxxx0xxx, ocf.zonebattery.charge=100 &
1085 ocf.zonebattery.lowbattery=false",
1086     "if zonestatus=xxxxxxxxxx1xxx, ocf.zonebattery.charge=100 &
1087 ocf.zonebattery.lowbattery=true",
1088
1089     "if zonestatus=xxxxxxxx0xxxx, ocf.zonestatus.zonestatusreports='none'",
1090     "if zonestatus=xxxxxxxx0lxxx, ocf.zonestatus.zonestatusreports='statuschangeonly'
1091 ",
1092     "if zonestatus=xxxxxxxx10xxx, ocf.zonestatus.zonestatusreports='alarmclearonly' ",
1093     "if zonestatus=xxxxxxxx1lxxx,
1094 ocf.zonestatus.zonestatusreports='statuschangeandalarmclear'",
1095
1096     "if zonestatus=xxxxxxx0xxxxxx, ocf.zonestatus.fault=false",
1097     "if zonestatus=xxxxxxx1xxxxxx, ocf.zonestatus.fault=true",
1098
1099     "if zonestatus=xxxxxx0xxxxxxx, ocf.zonepowersource.powerSources=['AC (Mains)
1100 Power'] & ocf.zonepowersource.sourcefault=false",
1101     "if zonestatus=xxxxxx1xxxxxxx, ocf.zonepowersource.powerSources=['AC (Mains)
1102 Power'] & ocf.zonepowersource.sourcefault=true",
1103
1104     "if zonestatus=xxxxx0xxxxxxx, ocf.zonestatus.test=false",
1105     "if zonestatus=xxxxx1xxxxxxx, ocf.zonestatus.test=true",
1106
1107     "if zonestatus=xxxx0xxxxxxx, ocf.zonepowersource.powerSources=['Internal
1108 Battery'] & oic.r.ias.zone.zonebattery.defect=false & oic.r.ias.zone.zonebattery.charge=100.",
1109     "if zonestatus=xxxx1xxxxxxx,
1110 oic.r.ias.zone.zonepowersource.powerSources=['Internal Battery'] &
1111 oic.r.ias.zone.zonebattery.defect=true & oic.r.ias.zone.zonebattery.charge=100."
1112 ],
1113 "x-from-ocf": [
1114     "N/A"
1115 ]
1116 },
1117 },
1118 "IAS_CIE_address": {

```

```

1119         "type": "string",
1120         "description": "Address of IAS Control and Indicating Equipment (CIE)",
1121         "x-ocf-conversion": {
1122             "x-ocf-alias": "oic.r.iaszone",
1123             "x-to-ocf": [
1124                 "ocf.iascieaddress= IAS_CIE_address"
1125             ],
1126             "x-from-ocf": [
1127                 "N/A"
1128             ]
1129         },
1130     },
1131     "zoneID": {
1132         "type": "integer",
1133         "description": "Unique id allocated by IAS CIE",
1134         "x-ocf-conversion": {
1135             "x-ocf-alias": "oic.r.iaszone",
1136             "x-to-ocf": [
1137                 "ocf.zoneid=zoneID"
1138             ],
1139             "x-from-ocf": [
1140                 "N/A"
1141             ]
1142         },
1143     },
1144     "numberofzonesensitivitylevelsupported": {
1145         "type": "integer",
1146         "description": "Total number of sensitivity levels supported by the IAS Zone",
1147         "x-ocf-conversion": {
1148             "x-ocf-alias": "oic.r.iaszone",
1149             "x-to-ocf": [
1150                 "ocf.numzonesensitivitylevel= numberofzonesensitivitylevelsupported"
1151             ],
1152             "x-from-ocf": [
1153                 "N/A"
1154             ]
1155         },
1156     },
1157     "currentzonesensitivitylevel": {
1158         "type": "integer",
1159         "description": "Sensitivity level of IAS Zone",
1160         "x-ocf-conversion": {
1161             "x-ocf-alias": "oic.r.iaszone",
1162             "x-to-ocf": [
1163                 "ocf.currentzonesensitivitylevel = currentzonesensitivitylevel"
1164             ],
1165             "x-from-ocf": [
1166                 "N/A"
1167             ]
1168         },
1169     },
1170 },
1171 },
1172 },
1173 "type": "object",
1174 "allOf": [
1175     {"$ref": "#/definitions/zcl.iaszone.info"}
1176 ],
1177 "required": [ "zonestate" ]
1178 }
1179

```

1180 8.11 Level Control Cluster - Control

1181 8.11.1 Derived model

1182 The derived model: "zcl.levelcontrol.control.moveto".

1183 8.11.2 Property definition

1184 Table 19 provides the detailed per Property mapping for "zcl.levelcontrol.control.moveto".

1185

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)

1186 Table 20 provides the details of the Properties that are part of "zcl.levelcontrol.control.moveto".

1187 **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

1188 **8.11.3 Derived model definition**

```

1189 {
1190   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.levelcontrol.control.json#",
1191   "$schema": "http://json-schema.org/draft-04/schema#",
1192   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1193   "title": "Level Control Cluster - Control",
1194   "definitions": {
1195     "zcl.levelcontrol.control.moveto": {
1196       "properties": {
1197         "level": {
1198           "type": "integer",
1199           "description": "Move to certain dimming value as fast as possible ",
1200           "x-ocf-conversion": {
1201             "x-ocf-alias": "oic.r.light.dimming",
1202             "x-from-ocf": [
1203               "level=ocf.dimmingSetting * 254 /100 , transitiontime=0",
1204               "zcl.command.levelcontrol::movetolevel(level,transitiontime)"
1205             ],
1206             "x-to-ocf": [
1207               "N/A"
1208             ]
1209           }
1210         }
1211       }
1212     },
1213   },
1214   "type": "object",
1215   "allOf": [
1216     {"$ref": "#/definitions/zcl.levelcontrol.control.movetolevel"}
1217   ]
1218 }
1219

```

1220 **8.12 Level Control Cluster - Information**

1221 **8.12.1 Derived model**

1222 The derived model: "zcl.levelcontrol.info".

1223 **8.12.2 Property definition**

1224 Table 21 provides the detailed per Property mapping for "zcl.levelcontrol.info".

1225

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

1226

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

1227

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

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

1228

8.12.3 Derived model definition

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

1259

1260

```

{
  "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"
            ]
          }
        }
      }
    }
  }
}

```

1261

8.13 Occupancy Sensing Cluster - Information

1262

8.13.1 Derived model

1263

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

1264

8.13.2 Property definition

1265

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

1266

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	
--	--	--	--

1267 Table 24 provides the details of the Properties that are part of "zcl.occupancysensing.info".

1268 **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

1269 **8.13.3 Derived model definition**

```

1270 {
1271   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.occupancysensing.info.json#",
1272   "$schema": "http://json-schema.org/draft-04/schema#",
1273   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1274   "title": "Occupancy Sensing Cluster - Information",
1275   "definitions": {
1276     "zcl.occupancysensing.info": {
1277       "type": "object",
1278       "properties": {
1279         "occupancy": {
1280           "type": "number",
1281           "description": "x is a variable. Data type of occupancy in Zigbee is 8 bitmap (xxxxxxx)
1282 while data type of value in OCF is boolean type i.e., true=occupied, false=unoccupied",
1283           "x-ocf-conversion": {
1284             "x-ocf-alias": "oic.r.sensor.presence",
1285             "x-to-ocf": [
1286               "if occupancy =xxxxxxx0, then ocf.value = false",
1287               "if occupancy =xxxxxxx1, then ocf.value = true"
1288             ],
1289             "x-from-ocf": [
1290               "N/A"
1291             ]
1292           }
1293         }
1294       }
1295     }
1296   },
1297   "type": "object",
1298   "allOf": [
1299     {"$ref": "#/definitions/zcl.occupancysensing.info"}
1300   ],
1301   "required": [ "occupancy" ]
1302 }
1303

```

1304 **8.14 On/Off Cluster - Control**

1305 **8.14.1 Derived model**

1306 The derived model: "zcl.onoff.control.off".

1307 The derived model: "zcl.onoff.control.on".

1308 **8.14.2 Property definition**

1309 Table 25 provides the detailed per Property mapping for "zcl.onoff.control.off".

1310

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().

1311

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

1312

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

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

1313

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

1314

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().

1315

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

1316

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

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

1317

8.14.3 Derived model definition

1318

```

1319 {
1320   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.onoff.control.json#",
1321   "$schema": "http://json-schema.org/draft-04/schema#",
1322   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1323   "title": "On/Off Cluster - Control",
1324   "definitions": {
1325     "zcl.onoff.control.on": {
1326       "properties": {
1327         "onoff": {
1328           "type": "boolean",
1329           "description": "Turn on the device",
1330           "x-ocf-conversion": {
1331             "x-ocf-alias": "oic.r.switch.binary",
1332             "x-from-ocf": [
1333               "if ocf.value = true, zcl.command.onoff::on()."
1334             ],
1335             "x-to-ocf": [
1336               "N/A"
1337             ]
1338           }
1339         }
1340       },
1341       "zcl.onoff.control.off": {
1342         "properties": {
1343           "onoff": {
1344             "type": "boolean",
1345             "description": "Turn off the device",
1346             "x-ocf-conversion": {
1347               "x-ocf-alias": "oic.r.switch.binary",
1348               "x-from-ocf": [
1349                 "if ocf.value = false, zcl.command.onoff::off()."

```



```

1350         ],
1351         "x-to-ocf": [
1352             "N/A"
1353         ]
1354     }
1355 }
1356 }
1357 }
1358 },
1359 "type": "object",
1360 "allOf": [
1361     {"$ref": "#/definitions/zcl.onoff.control.on"},
1362     {"$ref": "#/definitions/zcl.onoff.control.off"}
1363 ]
1364 }
1365

```

1366 8.15 On/off Cluster - Information

1367 8.15.1 Derived model

1368 The derived model: "zcl.onoff".

1369 8.15.2 Property definition

1370 Table 29 provides the detailed per Property mapping for "zcl.onoff".

1371 **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

1372 Table 30 provides the details of the Properties that are part of "zcl.onoff".

1373 **Table 30 – The Properties of "zcl.onoff".**

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

1374 8.15.3 Derived model definition

```

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

```

```

1396     }
1397   }
1398 }
1399 }
1400 },
1401 "type": "object",
1402 "allOf": [
1403   {"$ref": "#/definitions/zcl.onoff.info"}
1404 ],
1405 "required": [ "onoff" ]
1406 }
1407

```

1408 **8.16 Temperature Measurement Cluster - Information**

1409 **8.16.1 Derived model**

1410 The derived model: "zcl.temperaturemeasurement.info".

1411 **8.16.2 Property definition**

1412 Table 31 provides the detailed per Property mapping for "zcl.temperaturemeasurement.info".

1413 **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

1414 Table 32 provides the details of the Properties that are part of "zcl.temperaturemeasurement.info".

1415 **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

1416 **8.16.3 Derived model definition**

```

1417 {
1418   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.temperaturemeasurement.info.json#",
1419   "$schema": "http://json-schema.org/draft-04/schema#",
1420   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1421   "title": "Temperature Measurement Cluster - Information",
1422   "definitions": {
1423     "zcl.temperaturemeasurement.info": {
1424       "type": "object",
1425       "properties": {
1426         "MeasuredValue": {
1427           "type": "number",
1428           "description": "Measured value",

```

```

1429     "x-ocf-conversion": {
1430       "x-ocf-alias": "oic.r.temperature",
1431       "x-to-ocf": [
1432         "ocf.temperature = MeasuredValue/100",
1433         "units = C"
1434       ],
1435       "x-from-ocf": [
1436         "N/A"
1437       ]
1438     },
1439   },
1440   "Tolerance": {
1441     "type": "number",
1442     "description": "Magnitude of the possible error",
1443     "x-ocf-conversion": {
1444       "x-ocf-alias": "oic.r.temperature",
1445       "x-to-ocf": [
1446         "ocf.precision = Tolerance/100"
1447       ],
1448       "x-from-ocf": [
1449         "N/A"
1450       ]
1451     },
1452   },
1453   "MinMeasuredValue": {
1454     "type": "number",
1455     "description": "Minimum value of MeasuredValue",
1456     "x-ocf-conversion": {
1457       "x-ocf-alias": "oic.r.temperature",
1458       "x-to-ocf": [
1459         "ocf.range[0] = MinMeasuredValue/100"
1460       ],
1461       "x-from-ocf": [
1462         "N/A"
1463       ]
1464     },
1465   },
1466   "MaxMeasuredValue": {
1467     "type": "number",
1468     "description": "Maximum value of MeasuredValue",
1469     "x-ocf-conversion": {
1470       "x-ocf-alias": "oic.r.temperature",
1471       "x-to-ocf": [
1472         "ocf.range[1] = MaxMeasuredValue/100"
1473       ],
1474       "x-from-ocf": [
1475         "N/A"
1476       ]
1477     },
1478   }
1479 }
1480 }
1481 },
1482 "type": "object",
1483 "allOf": [
1484   {"$ref": "#/definitions/zcl.temperaturemeasurement.info"}
1485 ],
1486 "required": [ "MeasuredValue", "Tolerance", "MinMeasuredValue", "MaxMeasuredValue" ]
1487 }
1488

```

1489 8.17 Thermostat Cluster - Cool - Control

1490 8.17.1 Derived model

1491 The derived model: "zcl.thermostat_cool.control.setpointraiselower".

1492 8.17.2 Property definition

1493 Table 33 provides the detailed per Property mapping for
 1494 "zcl.thermostat_cool.control.setpointraiselower".

1495 **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)

1496 Table 34 provides the details of the Properties that are part of
 1497 "zcl.thermostat_cool.control.setpointraiselower".

1498 **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

1499 **8.17.3 Derived model definition**

```

1500 {
1501   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.thermostat_cool.control.json#",
1502   "$schema": "http://json-schema.org/draft-04/schema#",
1503   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1504   "title": "Thermostat Cluster - Cool - Control",
1505   "definitions": {
1506     "zcl.thermostat_cool.control.setpointraiselower": {
1507       "type": "object",
1508       "properties": {
1509         "amount": {
1510           "type": "number",
1511           "description": "Set the target temperature with cool mode. Mode=0x01 is set by Zigbee 3.0
1512 translator",
1513           "x-ocf-conversion": {
1514             "x-ocf-alias": "oic.r.temperature",
1515             "x-from-ocf": [
1516               "if ocf.temperature is updated, then amount= ocf.temperature*100.",
1517               "zcl.command.thermostat::setpointraiselower(mode, amount)"
1518             ],
1519             "x-to-ocf": [
1520               "N/A"
1521             ]
1522           }
1523         }
1524       }
1525     }
1526   },
1527   "type": "object",
1528   "allof": [
1529     {"$ref": "#/definitions/zcl.thermostat_cool.control.setpointraiselower"}
1530   ]
1531 }
1532 
```

1533 **8.18 Thermostat Cluster - Current Temperature - Information**

1534 **8.18.1 Derived model**

1535 The derived model: "zcl.thermostat_currenttemperature.info".

1536 **8.18.2 Property definition**

1537 Table 35 provides the detailed per Property mapping for "zcl.thermostat_currenttemperature.info".

1538 **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

1539 Table 36 provides the details of the Properties that are part of
 1540 "zcl.thermostat_currenttemperature.info".

1541 **Table 36 – The Properties of "zcl.thermostat_currenttemperature.info".**

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

1542 **8.18.3 Derived model definition**

```

1543 {
1544   "id":
1545   "http://openinterconnect.org/zigbeemapping/schemas/zcl.thermostat_currenttemperature.info.json#",
1546   "$schema": "http://json-schema.org/draft-04/schema#",
1547   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1548   "title": "Thermostat Cluster - Current Temperature - Information ",
1549   "definitions": {
1550     "zcl.thermostat_currenttemperature.info": {
1551       "type": "object",
1552       "properties": {
1553         "localtemperature": {
1554           "type": "number",
1555           "description": "current sensed temperature",
1556           "x-ocf-conversion": {
1557             "x-ocf-alias": "oic.r.temperature",
1558             "x-to-ocf": [
1559               "ocf.temperature=localtempearture/100",
1560               "units = C"
1561             ],
1562             "x-from-ocf": [
1563               "N/A"
1564             ]
1565           }
1566         }
1567       }
1568     }
1569   },
1570   "type": "object",
1571   "allOf": [
1572     {"$ref": "#/definitions/zcl.thermostat_currenttemperature.info"}
1573   ],
1574   "required": [ "localtempearture" ]
1575 }
1576
  
```

1577 **8.19 Thermostat Cluster - Heat - Control**

1578 **8.19.1 Derived model**

1579 The derived model: "zcl.thermostat_heat.control.setpointraiseLower".

1580 **8.19.2 Property definition**

1581 Table 37 provides the detailed per Property mapping for
 1582 "zcl.thermostat_heat.control.setpointraiseLower".

1583 **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)

1584 Table 38 provides the details of the Properties that are part of
 1585 "zcl.thermostat_heat.control.setpointraiseLower".

1586 **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

1587 **8.19.3 Derived model definition**

```

1588 {
1589   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.thermostat_heat.control.json#",
1590   "$schema": "http://json-schema.org/draft-04/schema#",
1591   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1592   "title": "Thermostat Cluster - Heat - Control",
1593   "definitions": {
1594     "zcl.thermostat_heat.control.setpointraiseLower": {
1595       "type": "object",
1596       "properties": {
1597         "amount": {
1598           "type": "number",
1599           "description": "Set the target temperature with heat mode. Mode=0x00 is set by Zigbee 3.0
1600 translator",
1601           "x-ocf-conversion": {
1602             "x-ocf-alias": "oic.r.temperature",
1603             "x-from-ocf": [
1604               "if ocf.temperature is updated, then amount= ocf.temperature*100.",
1605               "zcl.command.thermostat::setpointraiseLower(mode, amount)"
1606             ],
1607             "x-to-ocf": [
1608               "N/A"
1609             ]
1610           }
1611         }
1612       }
1613     },
1614     "type": "object",
1615     "allof": [
1616       {"$ref": "#/definitions/zcl.thermostat_heat.control.setpointraiseLower"}
1617     ]
1618   }
1619 }
1620

```

1621 **8.20 Window Covering Cluster - Configuration - Control**

1622 **8.20.1 Derived model**

1623 The derived model: "zcl.windowcovering_conf.control".

1624 **8.20.2 Property definition**

1625 Table 39 provides the detailed per Property mapping for "zcl.windowcovering_conf.control".

1626 **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)

1627 Table 40 provides the details of the Properties that are part of "zcl.windowcovering_conf.control".

1628 **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.
--	--	--	--

1629 **8.20.3 Derived model definition**

```

1630 {
1631   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_conf.control.json#",
1632   "$schema": "http://json-schema.org/draft-04/schema#",
1633   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1634   "title": "Window Covering Cluster - Configuration - Control",
1635   "definitions": {
1636     "zcl.windowcovering_conf.control": {
1637       "properties": {
1638         "mode": {
1639           "type": "integer",
1640           "description": "Set the mode. x is a variable. Data type of Mode in Zigbee is 8 bitmap
1641 (xxxxxxx) while data type of mode in OCF is array with 4 Boolean type items(i.e., [Reversed Motor
1642 Direction, Calibration Mode, Maintenance Mode, LED]). Reversed Motor Direction : 0 = motor
1643 direction is normal, 1 = motor direction is reversed. Calibration Mode : 0 = run in normal mode, 1
1644 = run in calibration mode. Maintenance Mode : 0 = motor is running normally, 1 = motor is running
1645 in maintenance mode. LED: 0 = LEDs are off, 1 = LEDs will display feedback.",
1646           "x-ocf-conversion": {
1647             "x-ocf-alias": "oic.r.windowcovering",
1648             "x-from-ocf": [
1649               "if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxxxxx0.",
1650               "if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxxxxx1.",
1651               "if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxxxxx0x.",
1652               "if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxxxxx1x.",
1653               "if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxxx0xx.",
1654               "if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxxx1xx.",
1655               "if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxx0xxx.",
1656               "if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxx1xxx.",
1657               "zcl.command.general::write(mode)"
1658             ],
1659             "x-to-ocf": [
1660               "N/A"
1661             ]
1662           }
1663         },
1664         "Velocity-Lift": {
1665           "type": "integer",
1666           "description": "Set velocity associated with Lifting the Window Covering (cm/sec).",
1667           "x-ocf-conversion": {
1668             "x-ocf-alias": "oic.r.windowcovering",
1669             "x-from-ocf": [
1670               "if ocf.liftvelocity is updated, Velocity-Lift = ocf.liftvelocity.",
1671               "zcl.command.general::write(Velocity-Lift)"
1672             ],
1673             "x-to-ocf": [
1674               "N/A"
1675             ]
1676           }
1677         }
1678       }
1679     }
1680   }

```



```

1678     "Acceleration Time-Lift": {
1679         "type": "integer",
1680         "description": "Set ramp up times to reaching the velocity setting (0.1sec).",
1681         "x-ocf-conversion": {
1682             "x-ocf-alias": "oic.r.windowcovering",
1683             "x-from-ocf": [
1684                 "if ocf.liftaccelerationtime is updated, Acceleration Time-
1685 Lift=ocf.liftaccelerationtime.",
1686                 "zcl.command.general::write(Acceleration Time-Lift)"
1687             ],
1688             "x-to-ocf": [
1689                 "N/A"
1690             ]
1691         }
1692     },
1693     "Deceleration Time-Lift": {
1694         "type": "integer",
1695         "description": "Set ramp down times associated with stoping the velocity setting
1696 (0.1sec).",
1697         "x-ocf-conversion": {
1698             "x-ocf-alias": "oic.r.windowcovering",
1699             "x-from-ocf": [
1700                 "if ocf.liftdecelerationtime is updated, Deceleration Time-
1701 Lift=ocf.liftdecelerationtime.",
1702                 "zcl.command.general::write(Deceleration Time-Lift)"
1703             ],
1704             "x-to-ocf": [
1705                 "N/A"
1706             ]
1707         }
1708     }
1709 }
1710 }
1711 },
1712 "type": "object",
1713 "allOf": [
1714     {"$ref": "#/definitions/zcl.windowcovering_conf.control"}
1715 ]
1716 }
1717

```

1718 8.21 Window Covering Cluster - Configuration - Information

1719 8.21.1 Derived model

1720 The derived model: "zcl.windowcovering_conf.info".

1721 8.21.2 Property definition

1722 Table 41 provides the detailed per Property mapping for "zcl.windowcovering_conf.info".

1723 **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 =xxxx1xxx, ocf.mode.ledfeedback = true	
Acceleration Time-Lift	oic.r.windowcovering	ocf.liftaccelerationtime= Acceleration Time-Lift	N/A

1724 Table 42 provides the details of the Properties that are part of "zcl.windowcovering_conf.info".

1725 **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).

1726 **8.21.3 Derived model definition**

```

1727 {
1728   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_conf.info.json#",
1729   "$schema": "http://json-schema.org/draft-04/schema#",
1730   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1731   "title": "Window Covering Cluster - Configuration - Information",
1732   "definitions": {
1733     "zcl.windowcovering_conf.info": {
1734       "type": "object",
1735       "properties": {
1736         "Windowcoveringtype": {
1737           "type": "string",
1738           "description": "Type of Window Covering(i.e., [Rollershade,RollerShade-2 Motor,
1739 RollerShade-Exterior, RollerShade-Exterior-2 Motor, Drapery, Awning, Shutter, Tilt Blind - Tilt
1740 Only, Tilt Blind â€ Lift and Tilt, Projector Screen])",
1741           "x-ocf-conversion": {
1742             "x-ocf-alias": "oic.r.windowcovering",
1743             "x-to-ocf": [
1744               "if WindowCoveringType=0x00, ocf.windowcoveringtype= Rollershade.",
1745               "if WindowCoveringType=0x01, ocf.windowcoveringtype= RollerShade-2 Motor.",
1746               "if WindowCoveringType=0x02, ocf.windowcoveringtype= RollerShade-Exterior.",
1747               "if WindowCoveringType=0x03, ocf.windowcoveringtype= RollerShade-Exterior-2 Motor.",
1748               "if WindowCoveringType=0x04, ocf.windowcoveringtype= Drapery.",
1749               "if WindowCoveringType=0x05, ocf.windowcoveringtype= Awning.",
1750               "if WindowCoveringType=0x06, ocf.windowcoveringtype= Shutter.",
1751               "if WindowCoveringType=0x07, ocf.windowcoveringtype= Tilt Blind - Tilt Only.",
1752               "if WindowCoveringType=0x08, ocf.windowcoveringtype= Tilt Blind â€ Lift and Tilt.",
1753               "if WindowCoveringType=0x09, ocf.windowcoveringtype= Projector Screen."
1754             ],
1755             "x-from-ocf": [
1756               "N/A"
1757             ]
1758           }
1759         },
1760         "Config/Status": {
1761           "type": "integer",
1762           "description": " x is a variable. Config/Status in Zigbee maps to configstatus in OCF.
1763 Data type of Config/Status in Zigbee is 8 bitmap (xxxxxxx) : bit 0 = Operational, bit 1 = Online,
1764 bit 2 = Reversal, bit 3 = Control-Lift, bit 4 = Control-Tilt, bit 5 = Encoder-Lift, bit 6 =
1765 Encoder-Tilt. Operational: This status bit defines if the Window Covering is operational. 0 = Not
1766 Operational, 1 = Operational. Online: This status bit defines if the Window Covering is enabled for
1767 transmitting over the ZigBee network. 0 = Not Online, 1 = Online. Reversal: This status bit
1768 identifies if the direction of rotation for the Window Covering has been reversed in order for
1769 Open/Up commands to match the physical installation condition. 0 = Commands are normal, 1 = Open/Up
1770 Commands have been reversed. Control Lift: This status bit identifies if the window covering
1771 supports Open Loop or Closed Loop Lift Control. 0 = Lift control is Open Loop, 1 = Lift control is
1772 Closed. Control Tilt: This status bit identifies if the window covering supports Open Loop or
1773 Closed Loop Tilt Control. 0 = Tilt control is Open Loop, 1 = Tilt control is Closed. Encoder Lift:
1774 This status bit identifies if a Closed Loop Controlled Window Covering is employing an encoder for
1775 positioning the height of the window covering. 0 = Timer Controlled, 1 = Encoder Controlled.
1776 Encoder Tilt: This status bit identifies if a Closed Loop Controlled Window Covering is employing
1777 an encoder for tilting the window covering. 0 = Timer Controlled, 1 = Encoder Controlled.",
1778           "x-ocf-conversion": {

```

```

1779     "x-ocf-alias": "oic.r.windowcovering",
1780     "x-to-ocf": [
1781         "if Config/Status =xxxxxxx0, ocf.configstatus.operational = false",
1782         "if Config/Status =xxxxxxx1, ocf.configstatus.operational = true",
1783         "if Config/Status =xxxxxx0x, ocf.configstatus.online = false",
1784         "if Config/Status =xxxxxx1x, ocf.configstatus.online = true",
1785         "if Config/Status =xxxxx0xx, ocf.configstatus.rotationdirection = 'normal'",
1786         "if Config/Status =xxxxx1xx, ocf.configstatus.rotationdirection = 'reversed'",
1787         "if Config/Status =xxxx0xxx, ocf.configstatus.controllift = 'openloop'",
1788         "if Config/Status =xxxx1xxx, ocf.configstatus.controllift = 'closedloop'",
1789         "if Config/Status =xxx0xxxx, ocf.configstatus.controltilt = 'openloop'",
1790         "if Config/Status =xxx1xxxx, ocf.configstatus.controltilt = 'closedloop'",
1791         "if Config/Status =xx0xxxxx, ocf.configstatus.closedloopliftcontrol = 'timer'",
1792         "if Config/Status =xx1xxxxx, ocf.configstatus.closedloopliftcontrol = 'encoder'",
1793         "if Config/Status =x0xxxxxx, ocf.configstatus.closedlooptiltcontrol = 'timer'",
1794         "if Config/Status =x1xxxxxx, ocf.configstatus.closedlooptiltcontrol = 'encoder'"
1795     ],
1796     "x-from-ocf": [
1797         "N/A"
1798     ]
1799 },
1800 },
1801 "Mode": {
1802     "type": "integer",
1803     "description": "x is a variable. Mode in Zigbee maps to mode in OCF. Data type of Mode in
1804 Zigbee is 8 bitmap (xxxxxxx) : bit 0 = Reversed Motor Direction, bit 1 = Calibration Mode, bit 2 =
1805 Maintenance Mode, bit 3 = LED. Reversed Motor Direction : 0 = motor direction is normal, 1 = motor
1806 direction is reversed. Calibration Mode : 0 = run in normal mode, 1 = run in calibration mode.
1807 Maintenance Mode : 0 = motor is running normally, 1 = motor is running in maintenance mode. LED: 0
1808 = LEDs are off, 1 = LEDs will display feedback.",
1809     "x-ocf-conversion": {
1810         "x-ocf-alias": "oic.r.windowcovering",
1811         "x-to-ocf": [
1812             "if Mode =xxxxxxx0, ocf.mode.motordirection = false",
1813             "if Mode =xxxxxxx1, ocf.mode.motordirection = true",
1814             "if Mode =xxxxxx0x, ocf.mode.calibration = false",
1815             "if Mode =xxxxxx1x, ocf.mode.calibration = true",
1816             "if Mode =xxxxx0xx, ocf.mode.maintenance = false",
1817             "if Mode =xxxxx1xx, ocf.mode.maintenance = true",
1818             "if Mode =xxxx0xxx, ocf.mode.ledfeedback = false",
1819             "if Mode =xxxx1xxx, ocf.mode.ledfeedback = true"
1820         ],
1821         "x-from-ocf": [
1822             "N/A"
1823         ]
1824     }
1825 },
1826 "Velocity-Lift": {
1827     "type": "integer",
1828     "description": "Velocity associated with Lifting the Window Covering (cm/sec).",
1829     "x-ocf-conversion": {
1830         "x-ocf-alias": "oic.r.windowcovering",
1831         "x-to-ocf": [
1832             "ocf.liftvelocity = Velocity-Lift"
1833         ],
1834         "x-from-ocf": [
1835             "N/A"
1836         ]
1837     }
1838 },
1839 "Acceleration Time-Lift": {
1840     "type": "integer",
1841     "description": "Ramp up times to reaching the velocity setting (0.1sec).",
1842     "x-ocf-conversion": {
1843         "x-ocf-alias": "oic.r.windowcovering",
1844         "x-to-ocf": [
1845             "ocf.liftaccelerationtime= Acceleration Time-Lift"
1846         ],
1847         "x-from-ocf": [
1848             "N/A"
1849         ]
1850     }

```

```

1850     }
1851   },
1852   "Deceleration Time-Lift": {
1853     "type": "integer",
1854     "description": "Ramp down times associated with stoping the velocity setting (0.1sec).",
1855     "x-ocf-conversion": {
1856       "x-ocf-alias": "oic.r.windowcovering",
1857       "x-to-ocf": [
1858         "ocf.liftdecelerationtime= Deceleration Time-Lift"
1859       ],
1860       "x-from-ocf": [
1861         "N/A"
1862       ]
1863     }
1864   }
1865 }
1866 }
1867 },
1868 "type": "object",
1869 "allOf": [
1870   {"$ref": "#/definitions/zcl.windowcovering_conf.info"}
1871 ],
1872 "required": [ "Windowcoveringtype", "Config/Status", "Mode" ]
1873 }
1874

```

1875 **8.22 Window Covering Cluster - Lift Percentage - Control**

1876 **8.22.1 Derived model**

1877 The derived model: "zcl.windowcovering_liftpercentage.control.gotoliftpercentage".

1878 **8.22.2 Property definition**

1879 Table 43 provides the detailed per Property mapping for
 1880 "zcl.windowcovering_liftpercentage.control.gotoliftpercentage".

1881 **Table 43 – The Property mapping for**
 1882 **"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)

1883 Table 44 provides the details of the Properties that are part of
 1884 "zcl.windowcovering_liftpercentage.control.gotoliftpercentage".

1885 **Table 44 – The Properties of**
 1886 **"zcl.windowcovering_liftpercentage.control.gotoliftpercentage".**

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

1887 **8.22.3 Derived model definition**

```

1888 {
1889   "id":
1890   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_liftpercentage.control.json#"
1891   ,
1892   "$schema": "http://json-schema.org/draft-04/schema#",
1893   "description" : "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",

```

```

1894     "title": "Window Covering Cluster - Lift Percentage - Control",
1895     "definitions": {
1896       "zcl.windowcovering_liftpercentage.control.gotoliftpercentage": {
1897         "properties": {
1898           "percentageliftvalue": {
1899             "type": "integer",
1900             "description": "Adjust the window at the percentage lift value.",
1901             "x-ocf-conversion": {
1902               "x-ocf-alias": "oic.r.openlevel",
1903               "x-from-ocf": [
1904                 "if ocf.openLevel is updated, percentage lift value = ocf.openLevel.",
1905                 "zcl.command.windowcovering::gotoliftpercentage(percentageliftvalue)"
1906               ],
1907               "x-to-ocf": [
1908                 "N/A"
1909               ]
1910             }
1911           }
1912         }
1913       }
1914     },
1915     "type": "object",
1916     "allOf": [
1917       {"$ref": "#/definitions/zcl.windowcovering_liftpercentage.control.gotoliftpercentage"}
1918     ]
1919   }
1920 }

```

1921 8.23 Window Covering Cluster - Lift Percentage - Information

1922 8.23.1 Derived model

1923 The derived model: "zcl.windowcovering_liftpercentage.info".

1924 8.23.2 Property definition

1925 Table 45 provides the detailed per Property mapping for "zcl.windowcovering_liftpercentage.info".

1926 **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

1927 Table 46 provides the details of the Properties that are part of
 1928 "zcl.windowcovering_liftpercentage.info".

1929 **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

1930 8.23.3 Derived model definition

```

1931 {
1932   "id":
1933   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_liftpercentage.info.json#",
1934   "$schema": "http://json-schema.org/draft-04/schema#",
1935   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1936   "title": "Window Covering Cluster - Lift Percentage - Information",
1937   "definitions": {
1938     "zcl.windowcovering_liftpercentage.info": {
1939       "type": "object",

```



```

1940     "properties": {
1941       "CurrentPositionLiftPercentage": {
1942         "type": "integer",
1943         "description": "Position as a percentage between InstalledOpenLimit-Lift and
1944 InstalledClosedLimit-Lift",
1945         "x-ocf-conversion": {
1946           "x-ocf-alias": "oic.r.openlevel",
1947           "x-to-ocf": [
1948             "ocf.openLevel= CurrentPositionLiftPercentage"
1949           ],
1950           "x-from-ocf": [
1951             "N/A"
1952           ]
1953         }
1954       }
1955     }
1956   },
1957 },
1958 "type": "object",
1959 "allOf": [
1960   {"$ref": "#/definitions/zcl.windowcovering_liftpercentage.info"}
1961 ],
1962 "required": ["CurrentPositionLiftPercentage"]
1963 }
1964

```

1965 **8.24 Window Covering Cluster - Lift Position - Control**

1966 **8.24.1 Derived model**

1967 The derived model: "zcl.windowcovering_liftposition.control.gotoliftvalue".

1968 **8.24.2 Property definition**

1969 Table 47 provides the detailed per Property mapping for
 1970 "zcl.windowcovering_liftposition.control.gotoliftvalue".

1971 **Table 47 – The Property mapping for**
 1972 **"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)

1973 Table 48 provides the details of the Properties that are part of
 1974 "zcl.windowcovering_liftposition.control.gotoliftvalue".

1975 **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.

1976 **8.24.3 Derived model definition**

```

1977 {
1978   "id":
1979   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_liftposition.control.json#",
1980   "$schema": "http://json-schema.org/draft-04/schema#",
1981   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1982   "title": "Window Covering Cluster - Lift Position - Control",
1983   "definitions": {
1984     "zcl.windowcovering_liftposition.control.gotoliftvalue": {

```

```

1985     "properties": {
1986       "liftvalue": {
1987         "type": "integer",
1988         "description": "Adjust the window at the lift value.",
1989         "x-ocf-conversion": {
1990           "x-ocf-alias": "oic.r.openlevel",
1991           "x-from-ocf": [
1992             "if ocf.openLevel is updated, lift value= ocf.openLevel.",
1993             "zcl.command.windowcovering::gotoliftvalue(liftvalue)"
1994           ],
1995           "x-to-ocf": [
1996             "N/A"
1997           ]
1998         }
1999       }
2000     }
2001   },
2002   "type": "object",
2003   "allOf": [
2004     { "$ref": "#/definitions/zcl.windowcovering_liftposition.control.gotoliftvalue" }
2005   ]
2006 }
2007
2008

```

2009 8.25 Window Covering Cluster - Lift Position - Information

2010 8.25.1 Derived model

2011 The derived model: "zcl.windowcovering_liftposition.info".

2012 8.25.2 Property definition

2013 Table 49 provides the detailed per Property mapping for "zcl.windowcovering_liftposition.info".

2014 **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

2015 Table 50 provides the details of the Properties that are part of
2016 "zcl.windowcovering_liftposition.info".

2017 **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)

2018 8.25.3 Derived model definition

```
2019 {
2020   "id":
2021   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_liftposition.info.json#",
2022   "$schema": "http://json-schema.org/draft-04/schema#",
2023   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2024   "title": "Window Covering Cluster - Lift Position - Information",
2025   "definitions": {
2026     "zcl.windowcovering_liftposition.info": {
2027       "type": "object",
2028       "properties": {
2029         "InstalledClosedLimit-Lift": {
2030           "type": "integer",
2031           "description": "Close limit for lifting the Window Covering (cm)",
2032           "x-ocf-conversion": {
2033             "x-ocf-alias": "oic.r.openlevel",
2034             "x-to-ocf": [
2035               "ocf.range[0]= InstalledClosedLimit-Lift"
2036             ],
2037             "x-from-ocf": [
2038               "N/A"
2039             ]
2040           }
2041         },
2042         "InstalledOpenLimit-Lift": {
2043           "type": "integer",
2044           "description": "Open limit for lifting the Window Covering (cm)",
2045           "x-ocf-conversion": {
2046             "x-ocf-alias": "oic.r.openlevel",
2047             "x-to-ocf": [
2048               "ocf.range[1]= InstalledOpenLimit-Lift"
2049             ],
2050             "x-from-ocf": [
2051               "N/A"
2052             ]
2053           }
2054         },
2055         "CurrentPosition-Lift": {
2056           "type": "integer",
2057           "description": "Position of Window Covering from the top of the shade (cm)",
2058           "x-ocf-conversion": {
2059             "x-ocf-alias": "oic.r.openlevel",
2060             "x-to-ocf": [
2061               "ocf.openLevel= CurrentPosition-Lift"
2062             ],
2063             "x-from-ocf": [
2064               "N/A"
2065             ]
2066           }
2067         }
2068       }
2069     }
2070   },
2071   "type": "object",
2072   "allOf": [
2073     {"$ref": "#/definitions/zcl.windowcovering_liftposition.info"}
2074   ],
2075   "required": [ "InstalledClosedLimit-Lift", "InstalledOpenLimit-Lift", "CurrentPosition-Lift" ]
2076 }
2077
```

2078 8.26 Window Covering Cluster - Tilt Percentage - Control

2079 8.26.1 Derived model

2080 The derived model: "zcl.windowcovering_tiltpercentage.control.gotiltpercentage".

2081 **8.26.2 Property definition**

2082 Table 51 provides the detailed per Property mapping for
 2083 "zcl.windowcovering_tiltpercentage.control.gototiltpercentage".

2084 **Table 51 – The Property mapping for**
 2085 **"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)

2086 Table 52 provides the details of the Properties that are part of
 2087 "zcl.windowcovering_tiltpercentage.control.gototiltpercentage".

2088 **Table 52 – The Properties of**
 2089 **"zcl.windowcovering_tiltpercentage.control.gototiltpercentage".**

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

2090 **8.26.3 Derived model definition**

```

2091 {
2092   "id":
2093   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_tiltpercentage.control.json#"
2094 ,
2095   "$schema": "http://json-schema.org/draft-04/schema#",
2096   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2097   "title": "Window Covering Cluster - Tilt Percentage - Control",
2098   "definitions": {
2099     "zcl.windowcovering_tiltpercentage.control.gototiltpercentage": {
2100       "properties": {
2101         "percentagetiltvalue": {
2102           "type": "integer",
2103           "description": "Adjust the window at the percentage tilt value.",
2104           "x-ocf-conversion": {
2105             "x-ocf-alias": "oic.r.openlevel",
2106             "x-from-ocf": [
2107               "if ocf.openLevel is updated, percentage tilt value = ocf.openLevel.",
2108               "zcl.command.windowcovering::gototiltpercentage(percentagetiltvalue)"
2109             ],
2110             "x-to-ocf": [
2111               "N/A"
2112             ]
2113           }
2114         }
2115       }
2116     }
2117   },
2118   "type": "object",
2119   "allOf": [
2120     {"$ref": "#/definitions/zcl.windowcovering_tiltpercentage.control.gototiltpercentage"}
2121   ]
2122 }
2123
  
```

2124 **8.27 Window Covering Cluster - Tilt Percentage - Information**

2125 **8.27.1 Derived model**

2126 The derived model: "zcl.windowcovering_tiltpercentage.info".

2127 **8.27.2 Property definition**

2128 Table 53 provides the detailed per Property mapping for "zcl.windowcovering_tiltpercentage.info".

2129 **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

2130 Table 54 provides the details of the Properties that are part of
2131 "zcl.windowcovering_tiltpercentage.info".

2132 **Table 54 – The Properties of "zcl.windowcovering_tiltpercentage.info".**

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

2133 **8.27.3 Derived model definition**

```

2134 {
2135   "id":
2136   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_tiltpercentage.info.json#",
2137   "$schema": "http://json-schema.org/draft-04/schema#",
2138   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2139   "title": "Window Covering Cluster - Tilt Percentage - Information",
2140   "definitions": {
2141     "zcl.windowcovering_tiltpercentage.info": {
2142       "type": "object",
2143       "properties": {
2144         "CurrentPositionTiltPercentage": {
2145           "type": "integer",
2146           "description": "Tilt position as a percentage",
2147           "x-ocf-conversion": {
2148             "x-ocf-alias": "oic.r.openlevel",
2149             "x-to-ocf": [
2150               "ocf.openlevel=CurrentPositionTiltPercentage"
2151             ],
2152             "x-from-ocf": [
2153               "N/A"
2154             ]
2155           }
2156         }
2157       }
2158     }
2159   },
2160   "type": "object",
2161   "allOf": [
2162     {"$ref": "#/definitions/zcl.windowcovering_tiltpercentage.info"}
2163   ],
2164   "required": ["CurrentPositionTiltPercentage"]
2165 }
2166

```

2167 **8.28 Window Covering Cluster - Tilt Position - Control**

2168 **8.28.1 Derived model**

2169 The derived model: "zcl.windowcovering_tiltposition.control.gototiltvalue".

2170 **8.28.2 Property definition**

2171 Table 55 provides the detailed per Property mapping for
 2172 "zcl.windowcovering_tiltposition.control.gototiltvalue".

2173 **Table 55 – The Property mapping for**
 2174 **"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)

2175 Table 56 provides the details of the Properties that are part of
 2176 "zcl.windowcovering_tiltposition.control.gototiltvalue".

2177 **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.

2178 **8.28.3 Derived model definition**

```

2179 {
2180   "id":
2181   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_tiltposition.control.json#",
2182   "$schema": "http://json-schema.org/draft-04/schema#",
2183   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2184   "title": "Window Covering Cluster - Tilt Position - Control",
2185   "definitions": {
2186     "zcl.windowcovering_tiltposition.control.gototiltvalue": {
2187       "properties": {
2188         "tiltvalue": {
2189           "type": "integer",
2190           "description": "Adjust the window at the tilt value.",
2191           "x-ocf-conversion": {
2192             "x-ocf-alias": "oic.r.openlevel",
2193             "x-from-ocf": [
2194               "if ocf.openLevel is updated, tiltvalue= ocf.openLevel.",
2195               "zb.command.windowcovering::gototiltvalue(tiltvalue)"
2196             ],
2197             "x-to-ocf": [
2198               "N/A"
2199             ]
2200           }
2201         }
2202       }
2203     },
2204   },
2205   "type": "object",
2206   "allof": [
2207     {"$ref": "#/definitions/zcl.windowcovering_tiltposition.control.gototiltvalue"}
2208   ]
2209 }
2210
  
```

2211 **8.29 Window Covering Cluster - Tilt Position - Information**

2212 **8.29.1 Derived model**

2213 The derived model: "zcl.windowcovering_tiltposition.info".

2214 **8.29.2 Property definition**

2215 Table 57 provides the detailed per Property mapping for "zcl.windowcovering_tiltposition.info".

2216 **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

2217 Table 58 provides the details of the Properties that are part of
2218 "zcl.windowcovering_tiltposition.info".

2219 **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)

2220 **8.29.3 Derived model definition**

```

2221 {
2222   "id":
2223   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_tiltposition.info.json#",
2224   "$schema": "http://json-schema.org/draft-04/schema#",
2225   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2226   "title": "Window Covering Cluster - Tilt Position - Information",
2227   "definitions": {
2228     "zcl.windowcovering_tiltposition.info": {
2229       "type": "object",
2230       "properties": {
2231         "InstalledOpenLimit-Tilt": {
2232           "type": "integer",
2233           "description": "Close limit for tilting the Window Covering (0.1 degree)",
2234           "x-ocf-conversion": {
2235             "x-ocf-alias": "oic.r.openlevel",
2236             "x-to-ocf": [
2237               "ocf.range[0] = InstalledClosedLimit-Tilt"
2238             ],
2239             "x-from-ocf": [
2240               "N/A"
2241             ]
2242           }
2243         },
2244         "InstalledOpenLimit-Tilt": {
2245           "type": "integer",
2246           "description": "Open limit for tilting the Window Covering (0.1 degree)",
2247           "x-ocf-conversion": {
2248             "x-ocf-alias": "oic.r.openlevel",
2249             "x-to-ocf": [
2250               "ocf.range[1]= InstalledOpenLimit-Tilt"
2251             ],
2252             "x-from-ocf": [
2253               "N/A"
2254             ]
2255           }
2256         }
2257       }
2258     }
2259   }

```

```

2256     },
2257     "CurrentPosition-Tilt ": {
2258       "type": "integer",
2259       "description": "Tilt position of Window Covering from open (0.1 degree)",
2260       "x-ocf-conversion": {
2261         "x-ocf-alias": "oic.r.openlevel",
2262         "x-to-ocf": [
2263           "ocf.openlevel= CurrentPosition-Tilt"
2264         ],
2265         "x-from-ocf": [
2266           "N/A"
2267         ]
2268       }
2269     }
2270   }
2271 },
2272 },
2273 "type": "object",
2274 "allOf": [
2275   {"$ref": "#/definitions/zcl.windowcovering_tiltposition.info"}
2276 ],
2277 "required": [ "InstalledClosedLimit-Tilt", "InstalledOpenLimit-Tilt", "CurrentPosition-Tilt" ]
2278 }
2279
2280

```