



OCF 2.3 – Zigbee Resource Mapping specification – BTG

18
19
20

Legal Disclaimer

21
22

23 THIS IS A DRAFT SPECIFICATION DOCUMENT ONLY AND HAS NOT BEEN ADOPTED BY THE
24 OPEN CONNECTIVITY FOUNDATION. THIS DRAFT DOCUMENT MAY NOT BE RELIED UPON
25 FOR ANY PURPOSE OTHER THAN REVIEW OF THE CURRENT STATE OF THE DEVELOPMENT
26 OF THIS DRAFT DOCUMENT. THE OPEN CONNECTIVITY FOUNDATION AND ITS MEMBERS
27 RESERVE THE RIGHT WITHOUT NOTICE TO YOU TO CHANGE ANY OR ALL PORTIONS
28 HEREOF, DELETE PORTIONS HEREOF, MAKE ADDITIONS HERETO, DISCARD THIS DRAFT
29 DOCUMENT IN ITS ENTIRETY OR OTHERWISE MODIFY THIS DRAFT DOCUMENT AT ANY
30 TIME. YOU SHOULD NOT AND MAY NOT RELY UPON THIS DRAFT DOCUMENT IN ANY WAY,
31 INCLUDING BUT NOT LIMITED TO THE DEVELOPMENT OF ANY PRODUCTS OR SERVICES.
32 IMPLEMENTATION OF THIS DRAFT DOCUMENT IS DONE AT YOUR OWN RISK AMEND AND
33 IT IS NOT SUBJECT TO ANY LICENSING GRANTS OR COMMITMENTS UNDER THE OPEN
34 CONNECTIVITY FOUNDATION INTELLECTUAL PROPERTY RIGHTS POLICY OR OTHERWISE.
35 IN CONSIDERATION OF THE OPEN CONNECTIVITY FOUNDATION GRANTING YOU ACCESS
36 TO THIS DRAFT DOCUMENT, YOU DO HEREBY WAIVE ANY AND ALL CLAIMS ASSOCIATED
37 HERewith INCLUDING BUT NOT LIMITED TO THOSE CLAIMS DISCUSSED BELOW, AS WELL
38 AS CLAIMS OF DETRIMENTAL RELIANCE.

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

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

42 Copying or other form of reproduction and/or distribution of these works are strictly prohibited.
43



44

45

CONTENTS

46	1	Scope.....	9
47	2	Normative references	9
48	3	Terms, definitions and symbols	9
49	3.1	Terms and definitions.....	9
50	3.2	Symbols and abbreviations	9
51	4	Document conventions and organization.....	9
52	4.1	Introduction.....	9
53	4.2	Conventions.....	10
54	4.3	Notation.....	10
55	4.4	Data types	10
56	5	Theory of Operation.....	11
57	5.1	Interworking Approach	11
58	5.1.1	General	11
59	5.1.2	Value Assignment.....	11
60	5.1.3	Property Naming.....	11
61	5.1.4	Range.....	11
62	5.1.5	Arrays.....	11
63	5.1.6	Default Mapping	11
64	5.1.7	Conditional Mapping	11
65	5.1.8	Method Invocation	11
66	6	Device Type Mapping	12
67	6.1	Introduction.....	12
68	6.2	Zigbee Device Types to OCF Device Types	12
69	7	Resource to ZigBee Cluster Equivalence	13
70	7.1	Introduction.....	13
71	7.2	Zigbee Clusters to OCF Resources	13
72	7.2.1	On/off	14
73	7.2.2	Level Control	14
74	7.2.3	Color Control	14
75	7.2.4	Thermostat	15
76	7.2.5	Window Covering.....	15
77	7.2.6	Temperature Measurement.....	16
78	7.2.7	Occupancy Sensing.....	16
79	7.2.8	IAS Zone	16
80	8	Detailed Mapping APIs	17
81	8.1	Color Control Cluster - Color Space - Control	17
82	8.1.1	Derived model	17
83	8.1.2	Property definition	17
84	8.1.3	Derived model definition	17
85	8.2	Color Control Cluster - Color Space - Information	18



86	8.2.1	Derived model	18
87	8.2.2	Property definition	18
88	8.2.3	Derived model definition	19
89	8.3	Color Control Cluster - Color Temperature - Control	19
90	8.3.1	Derived model	19
91	8.3.2	Property definition	19
92	8.3.3	Derived model definition	20
93	8.4	Color Control Cluster - Color Temperature - Information	20
94	8.4.1	Derived model	20
95	8.4.2	Property definition	20
96	8.4.3	Derived model definition	21
97	8.5	Color Control Cluster - Hue and Saturation - Control.....	22
98	8.5.1	Derived model	22
99	8.5.2	Property definition	22
100	8.5.3	Derived model definition	22
101	8.6	Color Control Cluster - Hue and Saturation - Information	23
102	8.6.1	Derived model	23
103	8.6.2	Property definition	23
104	8.6.3	Derived model definition	23
105	8.7	IAS Zone Cluster - Control.....	24
106	8.7.1	Derived model	24
107	8.7.2	Property definition	24
108	8.7.3	Derived model definition	24
109	8.8	IAS Zone Cluster - Information.....	25
110	8.8.1	Derived model	25
111	8.8.2	Property definition	25
112	8.8.3	Derived model definition	31
113	8.9	Level Control Cluster - Control.....	35
114	8.9.1	Derived model	35
115	8.9.2	Property definition	35
116	8.9.3	Derived model definition	35
117	8.10	Level Control Cluster - Information.....	35
118	8.10.1	Derived model	35
119	8.10.2	Property definition	36
120	8.10.3	Derived model definition	36
121	8.11	Occupancy Sensing Cluster - Information	36
122	8.11.1	Derived model	36
123	8.11.2	Property definition	36
124	8.11.3	Derived model definition	37
125	8.12	On/Off Cluster - Control	37
126	8.12.1	Derived model	37
127	8.12.2	Property definition	37
128	8.12.3	Derived model definition	38
129	8.13	On/off Cluster - Information.....	39



130	8.13.1	Derived model	39
131	8.13.2	Property definition	39
132	8.13.3	Derived model definition	39
133	8.14	Temperature Measurement Cluster - Information	39
134	8.14.1	Derived model	39
135	8.14.2	Property definition	40
136	8.14.3	Derived model definition	40
137	8.15	Thermostat Cluster - Cool - Control	41
138	8.15.1	Derived model	41
139	8.15.2	Property definition	41
140	8.15.3	Derived model definition	42
141	8.16	Thermostat Cluster - Current Temperature - Information	42
142	8.16.1	Derived model	42
143	8.16.2	Property definition	42
144	8.16.3	Derived model definition	42
145	8.17	Thermostat Cluster - Heat - Control	43
146	8.17.1	Derived model	43
147	8.17.2	Property definition	43
148	8.17.3	Derived model definition	43
149	8.18	Window Covering Cluster - Configuration - Control	44
150	8.18.1	Derived model	44
151	8.18.2	Property definition	44
152	8.18.3	Derived model definition	45
153	8.19	Window Covering Cluster - Configuration - Information	46
154	8.19.1	Derived model	46
155	8.19.2	Property definition	47
156	8.19.3	Derived model definition	50
157	8.20	Window Covering Cluster - Lift Percentage - Control.....	52
158	8.20.1	Derived model	52
159	8.20.2	Property definition	52
160	8.20.3	Derived model definition	53
161	8.21	Window Covering Cluster - Lift Percentage - Information	53
162	8.21.1	Derived model	53
163	8.21.2	Property definition	53
164	8.21.3	Derived model definition	53
165	8.22	Window Covering Cluster - Lift Position - Control.....	54
166	8.22.1	Derived model	54
167	8.22.2	Property definition	54
168	8.22.3	Derived model definition	54
169	8.23	Window Covering Cluster - Lift Position - Information.....	55
170	8.23.1	Derived model	55
171	8.23.2	Property definition	55
172	8.23.3	Derived model definition	55
173	8.24	Window Covering Cluster - Tilt Percentage - Control.....	56



174	8.24.1	Derived model	56
175	8.24.2	Property definition	56
176	8.24.3	Derived model definition	57
177	8.25	Window Covering Cluster - Tilt Percentage - Information	57
178	8.25.1	Derived model	57
179	8.25.2	Property definition	57
180	8.25.3	Derived model definition	57
181	8.26	Window Covering Cluster - Tilt Position - Control	58
182	8.26.1	Derived model	58
183	8.26.2	Property definition	58
184	8.26.3	Derived model definition	58
185	8.27	Window Covering Cluster - Tilt Position - Information.....	59
186	8.27.1	Derived model	59
187	8.27.2	Property definition	59
188	8.27.3	Derived model definition	59
189			
190			
191			



192

Figures

193

194

No table of figures entries found.



Tables

196	Table 1 Zigbee to OCF Device Type Mapping	12
197	Table 2 Zigbee Server Cluster to OCF Resource Type Mapping.....	13
198	Table 3 The property mapping for zcl.colorcontrol_csc.control.movetocolor.	17
199	Table 4 The properties of zcl.colorcontrol_csc.control.movetocolor.....	17
200	Table 5 The property mapping for zcl.colorcontrol_csc.info.....	18
201	Table 6 The properties of zcl.colorcontrol_csc.info.....	18
202	Table 7 The property mapping for zcl.colorcontrol_ct.control.movetocolortemperature.	19
203	Table 8 The properties of zcl.colorcontrol_ct.control.movetocolortemperature.....	20
204	Table 9 The property mapping for zcl.colorcontrol_ct.info.	20
205	Table 10 The properties of zcl.colorcontrol_ct.info.	21
206	Table 11 The property mapping for zcl.colorcontrol_hs.control.movetohueandsaturation.....	22
207	Table 12 The properties of zcl.colorcontrol_hs.control.movetohueandsaturation.	22
208	Table 13 The property mapping for zcl.colorcontrol_hs.info.	23
209	Table 14 The properties of zcl.colorcontrol_hs.info.	23
210	Table 15 The property mapping for zcl.iaszone.control.	24
211	Table 16 The properties of zcl.iaszone.control.	24
212	Table 17 The property mapping for zcl.iaszone.info.	25
213	Table 18 The properties of zcl.iaszone.info.	29
214	Table 19 The property mapping for zcl.levelcontrol.control.moveto.	35
215	Table 20 The properties of zcl.levelcontrol.control.moveto.	35
216	Table 21 The property mapping for zcl.levelcontrol.info.	36
217	Table 22 The properties of zcl.levelcontrol.info.	36
218	Table 23 The property mapping for zcl.occupancysensing.info.....	36
219	Table 24 The properties of zcl.occupancysensing.info.	37
220	Table 25 The property mapping for zcl.onoff.control.on.....	37
221	Table 26 The properties of zcl.onoff.control.on.	38
222	Table 27 The property mapping for zcl.onoff.control.off.....	38
223	Table 28 The properties of zcl.onoff.control.off.	38
224	Table 29 The property mapping for zcl.onoff.	39
225	Table 30 The properties of zcl.onoff.....	39
226	Table 31 The property mapping for zcl.temperaturemeasurement.info.	40
227	Table 32 The properties of zcl.temperaturemeasurement.info.	40
228	Table 33 The property mapping for zcl.thermostat_cool.control.setpointraiselower.	41
229	Table 34 The properties of zcl.thermostat_cool.control.setpointraiselower.	41
230	Table 35 The property mapping for zcl.thermostat_currenttemperature.info.	42
231	Table 36 The properties of zcl.thermostat_currenttemperature.info.	42
232	Table 37 The property mapping for zcl.thermostat_heat.control.setpointraiseLower.	43
233	Table 38 The properties of zcl.thermostat_heat.control.setpointraiseLower.....	43



234	Table 39 The property mapping for zcl.windowcovering_conf.control.	44
235	Table 40 The properties of zcl.windowcovering_conf.control.	44
236	Table 41 The property mapping for zcl.windowcovering_conf.info.	47
237	Table 42 The properties of zcl.windowcovering_conf.info.	48
238	Table 43 The property mapping for	
239	zcl.windowcovering_liftpercentage.control.gotoliftpercentage.	52
240	Table 44 The properties of zcl.windowcovering_liftpercentage.control.gotoliftpercentage.	52
241	Table 45 The property mapping for zcl.windowcovering_liftpercentage.info.	53
242	Table 46 The properties of zcl.windowcovering_liftpercentage.info.	53
243	Table 47 The property mapping for zcl.windowcovering_liftposition.control.gotoliftvalue.	54
244	Table 48 The properties of zcl.windowcovering_liftposition.control.gotoliftvalue.	54
245	Table 49 The property mapping for zcl.windowcovering_liftposition.info.	55
246	Table 50 The properties of zcl.windowcovering_liftposition.info.	55
247	Table 51 The property mapping for	
248	zcl.windowcovering_tiltpercentage.control.gototiltpercentage.	56
249	Table 52 The properties of zcl.windowcovering_tiltpercentage.control.gototiltpercentage.	57
250	Table 53 The property mapping for zcl.windowcovering_tiltpercentage.info.	57
251	Table 54 The properties of zcl.windowcovering_tiltpercentage.info.	57
252	Table 55 The property mapping for zcl.windowcovering_tiltposition.control.gototiltvalue.	58
253	Table 56 The properties of zcl.windowcovering_tiltposition.control.gototiltvalue.	58
254	Table 57 The property mapping for zcl.windowcovering_tiltposition.info.	59
255	Table 58 The properties of zcl.windowcovering_tiltposition.info.	59
256		
257		



258 **1 Scope**

259 The OCF Resource to Zigbee Cluster Mapping specification (“this specification”) provides detailed
260 mapping information between Zigbee defined Clusters and OCF defined Resources,

261 **2 Normative references**

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

265 OCF Core Specification, *Open Connectivity Foundation Core Specification, Version 2.0*

266 https://openconnectivity.org/specs/OCF_Core_Specification_v2.0.0.pdf

267 OCF Device Specification, *Open Connectivity Foundation Device Specification, Version 2.0*

268 https://openconnectivity.org/specs/OCF_Device_Specification_v2.0.0.pdf

269 OCF Resource Type Specification, *Open Connectivity Foundation Resource Type Specification,*
270 *Version 2.0*

271 https://openconnectivity.org/specs/OCF_Resource_Type_Specification_v2.0.0.pdf

272 OCF Bridging Specification, *Open Connectivity Foundation Bridging Specification, Version 2.0*

273 https://openconnectivity.org/specs/OCF_Bridging_Specification_v1.3.0.pdf

274 *Zigbee Cluster Library Specification, Version 1.0*

275 <http://www.zigbee.org/zigbee-for-developers/zigbee-3-0/>

276 ZigBee Lighting & Occupancy Device, Version 1.0

277 <http://www.zigbee.org/zigbee-for-developers/zigbee-3-0/>

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

279 [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)

280 [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)

281 IETF RFC 7159, *The JavaScript Object Notation (JSON) Data Interchange Format*, March 2014

282 <http://www.ietf.org/rfc/rfc7159.txt>

283 **3 Terms, definitions and symbols**

284 All terms and definitions as defined in the OCF Core Specification and OCF Bridging Specification
285 also apply to this specification.

286 **3.1 Terms and definitions**

287 None defined.

288 **3.2 Symbols and abbreviations**

289 None defined.

290 **4 Document conventions and organization**

291 **4.1 Introduction**

292 For the purposes of this document, the terms and definitions given in OCF Core Specification, OCF
293 Device Specification, and OCF Bridging Specification apply.



294 **4.2 Conventions**

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

299 **4.3 Notation**

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

302 Required (or shall or mandatory).

- 303 – These basic features shall be implemented to comply with OIC Core Architecture. The phrases
304 “shall not”, and “PROHIBITED” indicate behaviour that is prohibited, i.e. that if performed means
305 the implementation is not in compliance.

306 Recommended (or should).

- 307 – These features add functionality supported by OIC Core Architecture and should be
308 implemented. Recommended features take advantage of the capabilities OIC Core Architecture,
309 usually without imposing major increase of complexity. Notice that for compliance testing, if a
310 recommended feature is implemented, it shall meet the specified requirements to be in
311 compliance with these guidelines. Some recommended features could become requirements in
312 the future. The phrase “should not” indicates behaviour that is permitted but not recommended.

313 Allowed (or allowed).

- 314 – These features are neither required nor recommended by OIC Core Architecture, but if the
315 feature is implemented, it shall meet the specified requirements to be in compliance with these
316 guidelines.

- 317 – Conditionally allowed (CA)The definition or behaviour depends on a condition. If the specified
318 condition is met, then the definition or behaviour is allowed, otherwise it is not allowed.

319 Conditionally required (CR)

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

323 DEPRECATED

- 324 – Although these features are still described in this specification, they should not be implemented
325 except for backward compatibility. The occurrence of a deprecated feature during operation of
326 an implementation compliant with the current specification has no effect on the
327 implementation’s operation and does not produce any error conditions. Backward compatibility
328 may require that a feature is implemented and functions as specified but it shall never be used
329 by implementations compliant with this specification.

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

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

332 **4.4 Data types**

333 Data types are defined in the OCF Core Specification.



334 **5 Theory of Operation**

335 **5.1 Interworking Approach**

336 The interworking between ZigBee Clusters and OCF defined Resources is modelled using the
337 derived model syntax described in Derived Models for Interoperability Mapping Syntax

338 Within the defined syntax for derived modelling used by this Specification there are two blocks that
339 define the actual Property-Property equivalence or mapping. These blocks are identified by the
340 keywords 'x-to-ocf' and 'x-from-ocf'. Derived Models for Interoperability does not define a rigid
341 syntax for these blocks; they are free form string arrays that contain pseudo-coded mapping logic.

342 Within this specification we apply the rules in the following sub-sections to these blocks to ensure
343 consistency and re-usability and extensibility of the mapping logic that is defined.

344 **5.1.1 General**

345 All statements are terminated with a carriage return.

346 **5.1.2 Value Assignment**

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

349 **5.1.3 Property Naming**

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

353 The name of the OCF defined Property is prepended by the ecosystem designator 'ocf' to avoid
354 ambiguity (e.g. 'ocf.step')

355 **5.1.4 Range**

356 The range on the OCF side is fixed.

357 **5.1.5 Arrays**

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

360 **5.1.6 Default Mapping**

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

364 **5.1.7 Conditional Mapping**

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

366 If 'condition', then 'mapping'

367 is applied.

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

369 **5.1.8 Method Invocation**

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



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

376 6 Device Type Mapping

377 6.1 Introduction

378 This Section contains the mappings Zigbee Device Types to OCF Device Types.

379 6.2 Zigbee Device Types to OCF Device Types

380 Table 1 captures the equivalency mapping between Zigbee defined Device Types (Please see
381 Zigbee Cluster Library) and OCF defined Device Types (Please see OCF Device Specification).

382 **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

383 **7 Resource to ZigBee Cluster Equivalence**

384 **7.1 Introduction**

385 This clause introduces new resource types for resource mapping between Zigbee Clusters and
 386 OCF Resources and lists the complete set of applicable Zigbee Clusters and equivalent OCF
 387 Resource Type(s) in 7.2

388 **7.2 Zigbee Clusters to OCF Resources**

389 Table 2 captures the equivalency mapping between Zigbee defined Clusters (see Zigbee Cluster
 390 Library) and OCF defined Resource Types (see OCF Resource Type Specification). Detailed
 391 Property by Property mappings are provided in Section 7.1.

392 Clause 8 captures the mappings for mandatory server clusters for Zigbee 3.0 devices

393 **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) * 2 for lift (percentage scale and cm scale), 2 for tilt (percentage scale and cm scale)	oic. if.rw oic. if.a
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

394

395 **7.2.1 On/off**

396 The APIs with `zcl.onoff` define the mapping between an instance of an OCF Binary Switch Resource
397 and the Zigbee On/off Cluster. In Section 8.13, a RETRIEVE on an OCF Binary Switch Resource
398 maps to a general Read command on a Zigbee On/off Cluster. The value of Zigbee Attribute in
399 Zigbee On/off Cluster is retrieved via the general Read command and mapped with the value of
400 OCF Property in OCF Binary Switch Resource. In Section 8.12, an UPDATE on a Binary Switch
401 maps to a command invocation on either `on()` command or `off()` command of Zigbee On/off Cluster.
402 `value = true` maps to `on()`, `value = false` maps to `off()` of Zigbee On/off Cluster.

403

404 **7.2.2 Level Control**

405 The APIs with `zcl.levelcontrol` define the mapping between an instance of an OCF Dimming
406 Resource and the Zigbee Level Control Cluster. In Section 8.10, a RETRIEVE on an OCF Dimming
407 Resource maps to a general Read command on a Zigbee Level Control Cluster. The value of Zigbee
408 Attribute in Zigbee Level Control Cluster is retrieved via the general Read command and mapped
409 with the value of OCF Property in OCF Dimming Resource. In Section 8.9, an UPDATE on a
410 `dimmingsetting` maps to a command invocation on `movetolevel(level,transitiontime=0)` of Zigbee
411 Level Control Cluster.

412

413 **7.2.3 Color Control**

414 The APIs with `zcl.colorcontrol` define the mapping between instances of OCF Colour Resources
415 and the Zigbee Color Control Cluster. The OCF Colour Resources are OCF Hue and Saturation
416 Resource, OCF Colour Space Coordinate Resource, OCF Colour Temperature Resource.

417

418 The APIs with `zcl.colorcontrol_hs` define the mapping between an instance of OCF Hue and
419 Saturation Resources and the Zigbee Color Control Cluster. In Section 8.6, a RETRIEVE on an are
420 OCF Hue and Saturation Resource maps to a general Read command on a Zigbee Color Control
421 Cluster. The values of Zigbee Attributes in Zigbee Color Control Cluster are retrieved via the
422 general Read command and mapped with those of OCF Properties in OCF Hue and Saturation
423 Resource. In Section 8.5, an UPDATE on OCF Colour Hue and Saturation Resource maps to a
424 command invocation on `movetohueandsaturation(hue,saturation,transitiontime=0)` of Zigbee Color
425 Control Cluster.

426

427 The APIs with `zcl.colorcontrol_csc` define the mapping between an instance of OCF Colour Space
428 Coordinate Resource and the Zigbee Color Control Cluster. In Section 8.2, a RETRIEVE on an
429 OCF Colour Space Coordinate Resource maps to a general Read command on a Zigbee Color
430 Control Cluster. The values of Zigbee Attributes in Zigbee Color Control Cluster are retrieved via
431 the general Read command and mapped with those of OCF Properties in OCF Colour Space
432 Coordinate Resource. In Section 8.1, an UPDATE on OCF Colour Space Coordinate Resource
433 maps to a command invocation on `movetocolor(colorx,colory,transitiontime=0)` of Zigbee Color
434 Control Cluster.

435



436 The APIs with `zcl.colorcontrol_ct` define the mapping between an instance of OCF Colour
437 Temperature Resource and the Zigbee Color Control Cluster. In Section 8.4, a RETRIEVE on an
438 OCF Colour Temperature Resource maps to a general Read command on a Zigbee Color Control
439 Cluster. The values of Zigbee Attributes in Zigbee Color Control Cluster are retrieved via the
440 general Read command and mapped with those of OCF Properties in OCF Colour Temperature
441 Resource. In Section 8.3, an UPDATE on OCF Colour Temperature Resource maps to a command
442 invocation on `movetocolortemperature(colortemperature,transitiontime=0)` of Zigbee Color Control
443 Cluster.
444

445 **7.2.4 Thermostat**

446 The APIs with `zcl.thermostat` define the mapping between 3 instances of OCF Temperature
447 Resources and the Zigbee Thermostat Cluster. The 3 instances of OCF Temperature Resources
448 are for sensor, heater, cooler, respectively.
449

450 The API with `zcl.thermostat_currenttemperature` defines the mapping between an instance of OCF
451 Temperature Resource and the Zigbee Thermostat Cluster for sensor. In Section 8.16, a
452 RETRIEVE on an OCF Temperature Resource maps to a general Read command on a Zigbee
453 Thermostat Cluster. The value of Zigbee Attribute in Zigbee Thermostat Cluster is retrieved via the
454 general Read command and mapped with the value of OCF Property in OCF Temperature Resource.
455 The value represents the current temperature.
456

457 The API with `zcl.thermostat_heat` defines the mapping between an instance of OCF Temperature
458 Resource and the Zigbee Thermostat Cluster for heater. In Section 8.17, an Update on temperature
459 of OCF Temperature Resource maps to `setpointraiselower(mode=heat mode, amount)` on a Zigbee
460 Thermostat Cluster.
461

462 The API with `zcl.thermostat_cool` defines the mapping between an instance of OCF Temperature
463 Resource and the Zigbee Thermostat Cluster for cooler. In Section 8.15, an Update on temperature
464 of OCF Temperature Resource maps to `setpointraiselower(mode=cool mode, amount)` on a Zigbee
465 Thermostat Cluster.
466

467 **7.2.5 Window Covering**

468 The APIs with `zcl.windowcovering` define the mapping between 5 instances of OCF Resources and
469 the Zigbee Window Covering Cluster. The 5 instances of OCF Resources are the instance of OCF
470 Window Covering Resource and the 4 instances of OCF Open Level Resources. The 4 instances
471 of OCF Open Level Resources are for lift level with percentage scale, lift level with centimetre
472 scale, tilt level with percentage scale, tilt level with centimetre scale.
473

474 The API with `zcl.windowcovering_conf` defines the mapping between an instance of OCF Window
475 Covering Resource and the Zigbee window Covering Cluster. In Section 8.19, a RETRIEVE on an
476 OCF Window Covering Resource maps to a general Read command on a Zigbee Window Covering
477 Cluster. The values of Zigbee Attributes in Zigbee Window Covering Cluster is retrieved via the
478 general Read command and mapped with the value of OCF Property in OCF Window Covering
479 Resource. In Section 8.18, an Update on OCF Window Covering Resource maps to a general Write
480 command on a Zigbee Window Covering Cluster.
481

482 The API with `zcl.windowcovering_liftpercentage` defines the mapping between an instance of OCF
483 Open Level Resource and the Zigbee window Covering Cluster for lift with percentage scale. In
484 Section 8.21, a RETRIEVE on an OCF Open Level Resource maps to a general Read command
485 on `CurrentPositionLiftPercentage` of Zigbee Window Covering Cluster. The value of Zigbee
486 Attribute in Zigbee Window Covering Cluster is retrieved via the general Read command and
487 mapped with the value of OCF Property in OCF Open Level Resource. In Section 8.20, an Update



488 on OCF Open Level Resource maps to gotoliftpercentage(percentagevalue) on a Zigbee Window
489 Covering Cluster.

490
491 The API with zcl.windowcovering_liftposition defines the mapping between an instance of OCF
492 Open Level Resource and the Zigbee window Covering Cluster for lift with centimetre scale. In
493 Section 8.23, a RETRIEVE on an OCF Open Level Resource maps to a general Read command
494 on CurrentPosition-Lift of Zigbee Window Covering Cluster. The value of Zigbee Attribute in Zigbee
495 Window Covering Cluster is retrieved via the general Read command and mapped with the value
496 of OCF Property in OCF Open Level Resource. In Section 8.22, an Update on OCF Open Level
497 Resource maps to gotoliftvalue(liftvalue) on a Zigbee Window Covering Cluster.

498
499 The API with zcl.windowcovering_tiltpercentage defines the mapping between an instance of OCF
500 Open Level Resource and the Zigbee window Covering Cluster for tilt with percentage scale. In
501 Section 8.25, a RETRIEVE on an OCF Open Level Resource maps to a general Read command
502 on CurrentPositionTiltPercentage of Zigbee Window Covering Cluster. The value of Zigbee
503 Attribute in Zigbee Window Covering Cluster is retrieved via the general Read command and
504 mapped with the value of OCF Property in OCF Open Level Resource. In Section 8.24, an Update
505 on OCF Open Level Resource maps to gototiltpercentage(percentagevalue) on a Zigbee Window
506 Covering Cluster.

507
508 The API with zcl.windowcovering_tiltposition defines the mapping between an instance of OCF
509 Open Level Resource and the Zigbee window Covering Cluster for tilt with centimetre scale. In
510 Section 8.27, a RETRIEVE on an OCF Open Level Resource maps to a general Read command
511 on CurrentPosition-Tilt of Zigbee Window Covering Cluster. The value of Zigbee Attribute in Zigbee
512 Window Covering Cluster is retrieved via the general Read command and mapped with the value
513 of OCF Property in OCF Open Level Resource. In Section 8.26, an Update on OCF Open Level
514 Resource maps to gototiltvalue(tiltvalue) on a Zigbee Window Covering Cluster.

515

516 **7.2.6 Temperature Measurement**

517 The API with zcl.temperaturemeasurement defines the mapping between an instance of an OCF
518 Temperature Resource and the Zigbee Temperature Measurement Cluster for sensor. In Section
519 8.14, a RETRIEVE on an OCF Temperature Resource maps to a general Read command on a
520 Zigbee Temperature Measurement Cluster. The value of Zigbee Attribute in Zigbee Temperature
521 Measurement Cluster is retrieved via the general Read command and mapped with the value of
522 OCF Property in OCF Temperature Resource. The value represents the current temperature.

523 **7.2.7 Occupancy Sensing**

524 The API with defines the mapping between an instance of an OCF Presence Sensor Resource and
525 the Zigbee Occupancy Sensing Cluster. In Section 8.11, a RETRIEVE on an OCF Presence Sensor
526 Resource maps to a general Read command on a Zigbee Occupancy Sensing Cluster. The value
527 of Zigbee Attribute in Zigbee Occupancy Sensing Cluster is retrieved via the general Read
528 command and mapped with the value of OCF Property in OCF Presence Sensor.

529

530 **7.2.8 IAS Zone**

531 The API with defines the mapping between an instance of an OCF IAS Zone Resource and the
532 Zigbee IAS Zone Cluster. In Section 8.8, a RETRIEVE on an IAS Zone Resource maps to a general
533 Read command on a Zigbee IAS Zone Cluster. The values of Zigbee Attributes in Zigbee IAS Zone
534 Cluster are retrieved via the general Read command and mapped with those of OCF Properties in
535 OCF IAS Zone Resource. In Section 8.7, an Update on OCF IAS Zone Resource maps to a general
536 Write command on a Zigbee IAS Zone Cluster.



537 8 Detailed Mapping APIs

538 This section provides a mapping description (using JSON that aligns with the Derived Modelling
539 syntax described in [Derived Model White Paper]) for all Clusters and Resources that are within
540 scope

541 8.1 Color Control Cluster - Color Space - Control

542 8.1.1 Derived model

543 The derived model: zcl.colorcontrol_csc.control.movetocolor.

544 8.1.2 Property definition

545 **Table 3 The property mapping for zcl.colorcontrol_csc.control.movetocolor.**

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

546 **Table 4 The properties of zcl.colorcontrol_csc.control.movetocolor.**

Zigbee name	Property	Type	Required	Description
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.
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.

547 8.1.3 Derived model definition

```
548 {
549   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_csc.control.json#",
550   "$schema": "http://json-schema.org/draft-04/schema#",
551   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
552   "title": "Color Control Cluster - Color Space - Control",
553   "definitions": {
554     "zcl.colorcontrol_csc.control.movetocolor": {
555       "properties": {
556         "colorx": {
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         "colorx =ocf.csc[0]*65536 & transitiontime=0",
564         "zcl.command.colorcontrol::movetocolor(colorx,colory,transitiontime)."
565     ],
566     "x-to-ocf": [
567         "N/A"
568     ]
569 }
570 },
571 "colory": {
572     "type": "number",
573     "description": "Move to certain value(s) of color coordinates as fast as possible with
574 transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
575     "x-ocf-conversion": {
576         "x-ocf-alias": "oic.r.colour.csc",
577         "x-from-ocf": [
578             "colory= ocf.csc[1]*65536 & transitiontime=0",
579             "zcl.command.colorcontrol::movetocolor(colorx,colory,transitiontime)."
580         ],
581         "x-to-ocf": [
582             "N/A"
583         ]
584     }
585 }
586 }
587 }
588 },
589 "type": "object",
590 "allOf": [
591     {"$ref": "#/definitions/zcl.colorcontrol_csc.control.movetocolor"}
592 ]
593 }
594

```

595 8.2 Color Control Cluster - Color Space - Information

596 8.2.1 Derived model

597 The derived model: zcl.colorcontrol_csc.info.

598 8.2.2 Property definition

599 **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

600 **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



601 8.2.3 Derived model definition

```

602 {
603   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_csc.info.json#",
604   "$schema": "http://json-schema.org/draft-04/schema#",
605   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
606   "title": "Color Control Cluster - Color Space - Information",
607   "definitions": {
608     "zcl.colorcontrol_csc.info": {
609       "type": "object",
610       "properties": {
611         "currentX": {
612           "type": "integer",
613           "description": "current value of the normalized chromaticity value x, as defined in the
614 CIE xy Color Space",
615           "x-ocf-conversion": {
616             "x-ocf-alias": "oic.r.colour.csc",
617             "x-to-ocf": [
618               "ocf.csc[0] = currentX/65536"
619             ],
620             "x-from-ocf": [
621               "N/A"
622             ]
623           },
624         },
625         "currentY": {
626           "type": "integer",
627           "description": "current value of the normalized chromaticity value y, as defined in the
628 CIE xy Color Space",
629           "x-ocf-conversion": {
630             "x-ocf-alias": "oic.r.colour.csc",
631             "x-to-ocf": [
632               "ocf.csc[1] = currentY/65536"
633             ],
634             "x-from-ocf": [
635               "N/A"
636             ]
637           },
638         },
639       }
640     },
641   },
642   "type": "object",
643   "allOf": [
644     {"$ref": "#/definitions/zcl.colorcontrol_csc.info"}
645   ],
646   "required": ["currentx", "currenty"]
647 }
648

```

649 8.3 Color Control Cluster - Color Temperature - Control

650 8.3.1 Derived model

651 The derived model: zcl.colorcontrol_ct.control.movetocolortemperature.

652 8.3.2 Property definition

653 **Table 7 The property mapping for zcl.colorcontrol_ct.control.movetocolortemperature.**

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



654

Table 8 The properties of zcl.colorcontrol_ct.control.movetocolortemperature.

Zigbee name	Property	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.

655

8.3.3 Derived model definition

656

```

657 {
658   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_ct.control.json#",
659   "$schema": "http://json-schema.org/draft-04/schema#",
660   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
661   "title": "Color Control Cluster - Color Temperature - Control",
662   "definitions": {
663     "zcl.colorcontrol_ct.control.movetocolortemperature": {
664       "properties": {
665         "colortemperature": {
666           "type": "integer",
667           "description": "Move to certain value of colortemperature as fast as possible with
668 transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
669           "x-ocf-conversion": {
670             "x-ocf-alias": "oic.r.colour.colourtemperature",
671             "x-from-ocf": [
672               "colourtemperature=ocf.ct & transitiontime=0",
673               "zcl.command.colorcontrol::movetocolortemperature(colortemperature,transitiontime)"
674             ],
675             "x-to-ocf": [
676               "N/A"
677             ]
678           }
679         }
680       }
681     },
682     "type": "object",
683     "allOf": [
684       {"$ref": "#/definitions/zcl.colorcontrol_ct.control.movetocolortemperature"}
685     ]
686   }
687 }

```

688

8.4 Color Control Cluster - Color Temperature - Information

689

8.4.1 Derived model

690

The derived model: zcl.colorcontrol_ct.info.

691

8.4.2 Property definition

692

Table 9 The property mapping for zcl.colorcontrol_ct.info.

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



Table 10 The properties of zcl.colorcontrol_ct.info.

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

694 8.4.3 Derived model definition

```

695 {
696   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_ct.info.json#",
697   "$schema": "http://json-schema.org/draft-04/schema#",
698   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
699   "title": "Color Control Cluster - Color Temperature - Information",
700   "definitions": {
701     "zcl.colorcontrol_ct.info": {
702       "type": "object",
703       "properties": {
704         "colortemperaturemired": {
705           "type": "integer",
706           "description": "Scaled inverse of the current value of the color temperature",
707           "x-ocf-conversion": {
708             "x-ocf-alias": "oic.r.colour.colourtemperature",
709             "x-to-ocf": [
710               "ocf.ct = colortemperaturemired"
711             ],
712             "x-from-ocf": [
713               "N/A"
714             ]
715           }
716         },
717         "colortemphysicalmin": {
718           "type": "integer",
719           "description": "minimum mired value supported by the hardware",
720           "x-ocf-conversion": {
721             "x-ocf-alias": "oic.r.colour.colourtemperature",
722             "x-to-ocf": [
723               "ocf.range[0] = colortemphysicalmin"
724             ],
725             "x-from-ocf": [
726               "N/A"
727             ]
728           }
729         },
730         "colorphysicalmax": {
731           "type": "integer",
732           "description": "maximum mired value supported by the hardware",
733           "x-ocf-conversion": {
734             "x-ocf-alias": "oic.r.colour.colourtemperature",
735             "x-to-ocf": [
736               "ocf.range[1] = colorphysicalmax"
737             ],
738             "x-from-ocf": [
739               "N/A"
740             ]
741           }
742         }
743       }
744     }
745   },
746   "type": "object",

```



```

747     "allOf": [
748       { "$ref": "#/definitions/zcl.colorcontrol_ct.info" }
749     ],
750     "required": [ "colortemperatureired", "colortempphysicalmin", "colortempphysicalmax" ]
751   }
752

```

753 8.5 Color Control Cluster - Hue and Saturation - Control

754 8.5.1 Derived model

755 The derived model: zcl.colorcontrol_hs.control.movetohueandsaturation.

756 8.5.2 Property definition

757 **Table 11 The property mapping for zcl.colorcontrol_hs.control.movetohueandsaturation.**

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

758 **Table 12 The properties of zcl.colorcontrol_hs.control.movetohueandsaturation.**

Zigbee name	Property	Type	Required	Description
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.
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.

759 8.5.3 Derived model definition

```

760 {
761   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_hs.control.json#",
762   "$schema": "http://json-schema.org/draft-04/schema#",
763   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
764   "title": "Color Control Cluster - Hue and Saturation - Control",
765   "definitions": {
766     "zcl.colorcontrol_hs.control.movetohueandsaturation": {
767       "properties": {
768         "hue": {
769           "type": "integer",
770           "description": "Move to certain value(s) of hue or saturation or both as fast as possible
771 with transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
772           "x-ocf-conversion": {
773             "x-ocf-alias": "oic.r.colour.hs",
774             "x-from-ocf": [

```



```

775         "hue=ocf.hue/360 * 254 & transitiontime=0",
776         "zcl.command.colorcontrol::movetohueandsaturation(hue,saturation,transitiontime)"
777     ],
778     "x-to-ocf": [
779         "N/A"
780     ]
781 }
782 },
783 "saturation": {
784     "type": "integer",
785     "description": "Move to certain value(s) of hue or saturation or both as fast as possible
786 with transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
787     "x-ocf-conversion": {
788         "x-ocf-alias": "oic.r.colour.hs",
789         "x-from-ocf": [
790             "saturation=ocf.saturation & transitiontime=0",
791             "zcl.command.colorcontrol::movetohueandsaturation(hue,saturation,transitiontime)"
792         ],
793         "x-to-ocf": [
794             "N/A"
795         ]
796     }
797 }
798 }
799 }
800 },
801 "type": "object",
802 "allOf": [
803     {"$ref": "#/definitions/zcl.colorcontrol_hs.control.movetohueandsaturation"}
804 ]
805 }
806

```

807 8.6 Color Control Cluster - Hue and Saturation - Information

808 8.6.1 Derived model

809 The derived model: zcl.colorcontrol_hs.info.

810 8.6.2 Property definition

811 **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

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 8.6.3 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   "type": "object",
852   "allOf": [
853     { "$ref": "#/definitions/zcl.colorcontrol_hs.info" }
854   ],
855   "required": ["currenthue", "currentsaturation"]
856 }
857 }
858

```

859 8.7 IAS Zone Cluster - Control

860 8.7.1 Derived model

861 The derived model: zcl.iaszone.control.

862 8.7.2 Property definition

863 **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)

864 **Table 16 The properties of zcl.iaszone.control.**

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

865 8.7.3 Derived model definition

```

866 {
867   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.iaszone.control.json#",
868   "$schema": "http://json-schema.org/draft-04/schema#",
869   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
870   "title": "IAS Zone Cluster - Control",
871   "definitions": {

```




```

872 "zcl.iaszone.control": {
873   "properties": {
874     "currentzonesensitivitylevel": {
875       "type": "integer",
876       "description": "Set a sensitivity level of IAS Zone",
877       "x-ocf-conversion": {
878         "x-ocf-alias": "oic.r.ias.zone",
879         "x-from-ocf": [
880           "currentzonesensitivitylevel = ocf.currentzonesensitivitylevel",
881           "zcl.command.general::write(currentzonesensitivitylevel)"
882         ],
883         "x-to-ocf": [
884           "N/A"
885         ]
886       }
887     }
888   }
889 },
890 "type": "object",
891 "allOf": [
892   {"$ref": "#/definitions/zcl.iaszone.control"}
893 ]
894 }
895 }
896

```

8.8 IAS Zone Cluster - Information

8.8.1 Derived model

The derived model: zcl.iaszone.info.

8.8.2 Property definition

Table 17 The property mapping for zcl.iaszone.info.

Zigbee Property name	OCF Resource	To OCF	From OCF
IAS_CIE_address	oic.r.iaszone	ocf.iascieaddress= IAS_CIE_address	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 & zonestatus=xxxxxxxxxx1x, ocf.zonestatus.alarms=[''] if zonetype=0x000d & zonestatus=xxxxxxxxxx0, ocf.zonestatus.alarms=[''] if zonetype=0x000d & zonestatus=xxxxxxxxxx1, ocf.zonestatus.alarms=['intrusion'] if zonetype=0x000d & zonestatus=xxxxxxxxxx0x, ocf.zonestatus.alarms=[''] if zonetype=0x000d & zonestatus=xxxxxxxxxx1x, ocf.zonestatus.alarms=['presence'] if zonetype=0x000d & zonestatus=xxxxxxxxxx11,	N/A



	<pre> ocf.zonestatus.alarms=['intrusion','presence'] if zonetype=0x0015 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[''] if zonetype=0x0015 & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['1stportalopenclose'] if zonetype=0x0015 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[''] if zonetype=0x0015 & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['2ndportalopenclose'] if zonetype=0x0015 & zonestatus=xxxxxxxxxxxx11, ocf.zonestatus.alarms=['1stportalopenclose','2 ndportalopenclose'] if zonetype=0x0028 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[''] if zonetype=0x0028 & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['fire'] if zonetype=0x0028 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[''] if zonetype=0x0028 & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=[''] if zonetype=0x002a & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[''] if zonetype=0x002a & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['wateroverflow'] if zonetype=0x002a & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[''] if zonetype=0x002a & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=[''] if zonetype=0x002b & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[''] if zonetype=0x002b & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['CO'] if zonetype=0x002b & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[''] if zonetype=0x002b & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['cooking'] if zonetype=0x002b & zonestatus=xxxxxxxxxxxx11, ocf.zonestatus.alarms=['CO','cooking'] if zonetype=0x002c & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[''] if zonetype=0x002c & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['fall'] if zonetype=0x002c & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[''] if zonetype=0x002c & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['emergencybutton'] if zonetype=0x002c & </pre>
--	---



	<pre> zonestatus=xxxxxxxxxxx1, 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=xxxxxxxxxxx1, 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=xxxxxxxxxxx1, 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=xxxxxxxxxxx1, ocf.zonestatus.alarms=['panic','emergency'] if zonetype=0x021d & zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=[''] if zonetype=0x021d & zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=['panic'] if zonetype=0x021d & zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=[''] if zonetype=0x021d & zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=['emergency'] if zonetype=0x021d & zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=['panic','emergency'] if zonetype=0x0225 & zonestatus=xxxxxxxxxxx0, </pre>
--	---



		<pre>ocf.zonestatus.alarms=[''] if zonetype=0x0225 & zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=['glassbreak'] if zonetype=0x0225 & zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=[''] if zonetype=0x0225 & zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=[''] if zonetype=0x0226 & zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=[''] if zonetype=0x0226 & zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=[''] if zonetype=0x0226 & zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=[''] if zonetype=0x0226 & zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=[''] if zonetype=0x0229 & zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=[''] if zonetype=0x0229 & zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=[''] if zonetype=0x0229 & zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=[''] if zonetype=0x0229 & zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=[''] if zonetype=0xffff & zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=[''] if zonetype=0xffff & zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=[''] if zonetype=0xffff & zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=[''] if zonetype=0xffff & zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=[''] if zonestatus=xxxxxxxxxxx0xx, ocf.zonestatus.tamper=falseif zonestatus=xxxxxxxxxxx1xx, ocf.zonestatus.tamper=trueif zonestatus=xxxxxxxxxxx0xxx, ocf.zonebattery.charge=100 & ocf.zonebattery.lowbattery=falseif zonestatus=xxxxxxxxxxx1xxx, ocf.zonebattery.charge=100 & ocf.zonebattery.lowbattery=trueif zonestatus=xxxxxxxx00xxxx, ocf.zonestatus.zonestatusreports='none'if zonestatus=xxxxxxxx01xxxx, ocf.zonestatus.zonestatusreports='statuschang eonly' if zonestatus=xxxxxxxx10xxxx, ocf.zonestatus.zonestatusreports='alarmclearo nly' if zonestatus=xxxxxxxx11xxxx, ocf.zonestatus.zonestatusreports='statuschang eandalarmclear'if zonestatus=xxxxxxxx0xxxxx, ocf.zonestatus.fault=falseif zonestatus=xxxxxxxx1xxxxx, ocf.zonestatus.fault=trueif zonestatus=xxxxxxxx0xxxxx, ocf.zonepowersource.powerSources=['AC</pre>
--	--	---



		(Mains) Power'] & ocf.zonepowersource.sourcefault=falseif zonestatus=xxxxx1xxxxxx, ocf.zonepowersource.powerSources=[' AC (Mains) Power'] & ocf.zonepowersource.sourcefault=trueif zonestatus=xxxx0xxxxxx, ocf.zonestatus.test=falseif zonestatus=xxxx1xxxxxx, ocf.zonestatus.test=trueif zonestatus=xxx0xxxxxx, ocf.zonepowersource.powerSources=['Internal Battery'] & oic.r.ias.zone.zonebattery.defect=false & oic.r.ias.zone.zonebattery.charge=100. if zonestatus=xxx1xxxxxx, oic.r.ias.zone.zonepowersource.powerSources =['Internal Battery'] & oic.r.ias.zone.zonebattery.defect=true & oic.r.ias.zone.zonebattery.charge=100.	
zonestate	oic.r.iaszone	if zonestate=0x00, ocf.zonestate=falseif zonestate=0x01, ocf.zonestate=true	N/A
zoneID	oic.r.iaszone	ocf.zoneid=zoneID	N/A
numberofzonesensitivitylevels supported	oic.r.iaszone	ocf.numzonesensitivitylevel= numberofzonesensitivitylevels supported	N/A
currentzonesensitivitylevel	oic.r.iaszone	ocf.currentzonesensitivitylevel currentzonesensitivitylevel	= N/A
zonetype	oic.r.iaszone	if zonetype=0x0000, ocf.zonetype=Standard CIEif zonetype=0x000d, ocf.zonetype=Motion sensorif zonetype=0x0015, ocf.zonetype=Contact switchif zonetype=0x0028, ocf.zonetype=Fire sensorif zonetype=0x002a, ocf.zonetype=Water sensorif zonetype=0x002b, ocf.zonetype=Carbon Monoxide (CO) sensorif zonetype=0x002c, ocf.zonetype=Personal emergency deviceif zonetype=0x002d, ocf.zonetype=Vibration/Movement sensorif zonetype=0x010f, ocf.zonetype=Remote Controlif zonetype=0x0115, ocf.zonetype=Key fobif zonetype=0x021d, ocf.zonetype=Keypadif zonetype=0x0225, ocf.zonetype=Standard Warning Deviceif zonetype=0x0226, ocf.zonetype=Glass break sensorif zonetype=0x0229, ocf.zonetype=Security repeaterif zonetype=0xffff, ocf.zonetype=Invalid Zone Type	N/A

Table 18 The properties of zcl.iaszone.info.

902

Zigbee Property name	Type	Required	Description
IAS_CIE_address	string	no	Address of IAS Control and Indicating Equipment (CIE)



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.
zonestate	boolean	yes	Enrollment status of IAS Zone false=not enrolled, true=enrolled
zoneID	integer	no	Unique id allocated by IAS CIE



numberofzonesensitivitylevelsupported	integer	no	Total number of sensitivity levels supported by the IAS Zone
currentzonesensitivitylevel	integer	no	Sensitivity level of IAS Zone
zonetype	string	no	Zonetype and Meaning of Alarm1 and Alarm2 zonestatus

903 8.8.3 Derived model definition

```
904 {
905   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.iaszone.info.json#",
906   "$schema": "http://json-schema.org/draft-04/schema#",
907   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
908   "title": "IAS Zone Cluster - Information",
909   "definitions": {
910     "zcl.iaszone.info": {
911       "type": "object",
912       "properties": {
913         "zonestate": {
914           "type": "boolean",
915           "description": "Enrollment status of IAS Zone false=not enrolled, true=enrolled",
916           "x-ocf-conversion": {
917             "x-ocf-alias": "oic.r.iaszone",
918             "x-to-ocf": [
919               "if zonestate=0x00, ocf.zonestate=false",
920               "if zonestate=0x01, ocf.zonestate=true"
921             ],
922             "x-from-ocf": [
923               "N/A"
924             ]
925           }
926         },
927         "zonetype": {
928           "type": "string",
929           "description": "Zonetype and Meaning of Alarm1 and Alarm2 zonestatus",
930           "x-ocf-conversion": {
931             "x-ocf-alias": "oic.r.iaszone",
932             "x-to-ocf": [
933               "if zonetype=0x0000, ocf.zonetype=Standard CIE",
934               "if zonetype=0x000d, ocf.zonetype=Motion sensor",
935               "if zonetype=0x0015, ocf.zonetype=Contact switch",
936               "if zonetype=0x0028, ocf.zonetype=Fire sensor",
937               "if zonetype=0x002a, ocf.zonetype=Water sensor",
938               "if zonetype=0x002b, ocf.zonetype=Carbon Monoxide (CO) sensor",
939               "if zonetype=0x002c, ocf.zonetype=Personal emergency device",
940               "if zonetype=0x002d, ocf.zonetype=Vibration/Movement sensor",
941               "if zonetype=0x010f, ocf.zonetype=Remote Control",
942               "if zonetype=0x0115, ocf.zonetype=Key fob",
943               "if zonetype=0x021d, ocf.zonetype=Keypad",
944               "if zonetype=0x0225, ocf.zonetype=Standard Warning Device",
945               "if zonetype=0x0226, ocf.zonetype=Glass break sensor",
946               "if zonetype=0x0229, ocf.zonetype=Security repeater",
947               "if zonetype=0xffff, ocf.zonetype=Invalid Zone Type"
948             ],
949             "x-from-ocf": [
950               "N/A"
951             ]
952           }
953         }
954       },
955       "zonestatus": {
956         "type": "array",
957         "items": {
958           "type": "integer"
959         },
960         "description": "x is a variable. zonestatus in Zigbee maps to zonestatus, zonebattery,
```



```

960 and zonepowersource in OCF. Data type of zonestatus in Zigbee is 16 bitmap (xxxxxxxxxxxxxxxx) : bit
961 0 = Alarm1, bit 1 = Alarm2, bit 2 = Tamper, bit 3 = Battery, bit 4 = Supervision reports, bit 5 =
962 Restore reports, bit 6 = Trouble, bit 7 = AC (mains), bit 8 = Test, bit 9 = Battery Defect. Alarm1 :
963 1 = opened or alarmed 0 = closed or not alarmed, Alarm2 : 1 = opened or alarmed 0 = closed or not
964 alarmed, Tamper : 1 = Tampered 0 = Not tampered, Battery : 1 = Low battery 0 = Battery OK,
965 Supervision reports : 1 = Reports 0 = Does not report, Restore reports : 1 = Reports restore 0 =
966 Does not report restore, Trouble : 1 = Trouble/Failure 0 = OK, AC (mains) : 1 = AC/Mains fault 0 =
967 AC/Mains OK, Test : 1 = Sensor is in test mode 0 = Sensor is in operation mode, Battery Defect : 1 =
968 Sensor detects a defective battery 0 = Sensor battery is functioning.",
969 "x-ocf-conversion": {
970   "x-ocf-alias": "oic.r.iaszone",
971   "x-to-ocf": [
972     "if zonetype=0x0000 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
973     "if zonetype=0x0000 & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['system']",
974     "if zonetype=0x0000 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
975     "if zonetype=0x0000 & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['']",
976     "if zonetype=0x000d & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
977     "if zonetype=0x000d & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['intrusion']",
978     "if zonetype=0x000d & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
979     "if zonetype=0x000d & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['presence']",
980     "if zonetype=0x000d & zonestatus=xxxxxxxxxxxx11, ocf.zonestatus.alarms=['intrusion','presence']",
981     "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['1stportalopenclose']",
982     "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['2ndportalopenclose']",
983     "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['1stportalopenclose','2ndportalopenclose']",
984     "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['1stportalopenclose','2ndportalopenclose']",
985     "if zonetype=0x0028 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['fire']",
986     "if zonetype=0x0028 & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['fire']",
987     "if zonetype=0x0028 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['fire']",
988     "if zonetype=0x0028 & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['fire']",
989     "if zonetype=0x002a & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['wateroverflow']",
990     "if zonetype=0x002a & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['wateroverflow']",
991     "if zonetype=0x002a & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['wateroverflow']",
992     "if zonetype=0x002a & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['wateroverflow']",
993     "if zonetype=0x002b & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['CO']",
994     "if zonetype=0x002b & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['CO']",
995     "if zonetype=0x002b & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['CO','cooking']",
996     "if zonetype=0x002b & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['CO','cooking']",
997     "if zonetype=0x002c & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['fall','emergencybutton']",
998     "if zonetype=0x002c & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['fall','emergencybutton']",
999     "if zonetype=0x002c & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['fall','emergencybutton']",
1000    "if zonetype=0x002c & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['fall','emergencybutton']",
1001    "if zonetype=0x002d & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['movement','vibration']",
1002    "if zonetype=0x002d & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['movement','vibration']",
1003    "if zonetype=0x002d & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['movement','vibration']",
1004    "if zonetype=0x002d & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['movement','vibration']",
1005    "if zonetype=0x002e & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['vibration','movement']",
1006    "if zonetype=0x002e & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['vibration','movement']",
1007    "if zonetype=0x002e & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['vibration','movement']",
1008    "if zonetype=0x002e & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['vibration','movement']"
1009  ],
1010 }
1011 }
1012 }
1013 }
1014 }
1015 }
1016 }
1017 }
1018 }
1019 }
1020 }
1021 }
1022 }
1023 }
1024 }
1025 }
1026 }
1027 }
1028 }
1029 }
1030 }

```




```
1031         "if zonetype=0x010f & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1032         "if zonetype=0x010f & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['panic']",
1033         "if zonetype=0x010f & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1034         "if zonetype=0x010f & zonestatus=xxxxxxxxxxxxx1x,
1035 ocf.zonestatus.alarms=['emergency']",
1036         "if zonetype=0x010f & zonestatus=xxxxxxxxxxxxx11,
1037 ocf.zonestatus.alarms=['panic', 'emergency']",
1038
1039         "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1040         "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['panic']",
1041         "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1042         "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxxx1x,
1043 ocf.zonestatus.alarms=['emergency']",
1044         "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxxx11,
1045 ocf.zonestatus.alarms=['panic', 'emergency']",
1046
1047         "if zonetype=0x021d & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1048         "if zonetype=0x021d & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['panic']",
1049         "if zonetype=0x021d & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1050         "if zonetype=0x021d & zonestatus=xxxxxxxxxxxxx1x,
1051 ocf.zonestatus.alarms=['emergency']",
1052         "if zonetype=0x021d & zonestatus=xxxxxxxxxxxxx11,
1053 ocf.zonestatus.alarms=['panic', 'emergency']",
1054
1055         "if zonetype=0x0225 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1056         "if zonetype=0x0225 & zonestatus=xxxxxxxxxxxxx1,
1057 ocf.zonestatus.alarms=['glassbreak']",
1058         "if zonetype=0x0225 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1059         "if zonetype=0x0225 & zonestatus=xxxxxxxxxxxxx1x, ocf.zonestatus.alarms=[' ']",
1060
1061         "if zonetype=0x0226 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1062         "if zonetype=0x0226 & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=[' ']",
1063         "if zonetype=0x0226 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1064         "if zonetype=0x0226 & zonestatus=xxxxxxxxxxxxx1x, ocf.zonestatus.alarms=[' ']",
1065
1066         "if zonetype=0x0229 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1067         "if zonetype=0x0229 & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=[' ']",
1068         "if zonetype=0x0229 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1069         "if zonetype=0x0229 & zonestatus=xxxxxxxxxxxxx1x, ocf.zonestatus.alarms=[' ']",
1070
1071         "if zonetype=0xffff & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1072         "if zonetype=0xffff & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=[' ']",
1073         "if zonetype=0xffff & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1074         "if zonetype=0xffff & zonestatus=xxxxxxxxxxxxx1x, ocf.zonestatus.alarms=[' ']",
1075
1076         "if zonestatus=xxxxxxxxxx0xx, ocf.zonestatus.tamper=false",
1077         "if zonestatus=xxxxxxxxxx1xx, ocf.zonestatus.tamper=true",
1078
1079         "if zonestatus=xxxxxxxx0xxx, ocf.zonebattery.charge=100 &
1080 ocf.zonebattery.lowbattery=false",
1081         "if zonestatus=xxxxxxxx1xxx, ocf.zonebattery.charge=100 &
1082 ocf.zonebattery.lowbattery=true",
1083
1084         "if zonestatus=xxxxxxx00xxxx, ocf.zonestatus.zonestatusreports='none'",
1085         "if zonestatus=xxxxxxx01xxxx, ocf.zonestatus.zonestatusreports='statuschangeonly'
1086 ",
1087         "if zonestatus=xxxxxxx10xxxx, ocf.zonestatus.zonestatusreports='alarmclearonly' ",
1088         "if zonestatus=xxxxxxx11xxxx,
1089 ocf.zonestatus.zonestatusreports='statuschangeandalarmclear' ",
1090
1091         "if zonestatus=xxxxxxx0xxxxxx, ocf.zonestatus.fault=false",
1092         "if zonestatus=xxxxxxx1xxxxxx, ocf.zonestatus.fault=true",
1093
1094         "if zonestatus=xxxxxx0xxxxxxx, ocf.zonepowersource.powerSources=['AC (Mains) Power']
1095 & ocf.zonepowersource.sourcefault=false",
1096         "if zonestatus=xxxxxx1xxxxxxx, ocf.zonepowersource.powerSources=['AC (Mains) Power']
1097 & ocf.zonepowersource.sourcefault=true",
1098
1099         "if zonestatus=xxxxx0xxxxxxx, ocf.zonestatus.test=false",
1100         "if zonestatus=xxxxx1xxxxxxx, ocf.zonestatus.test=true",
1101
```



```
1102         "if zonestatus=xxxx0xxxxxxxx, ocf.zonepowersource.powerSources=['Internal Battery']
1103 & oic.r.ias.zone.zonebattery.defect=false & oic.r.ias.zone.zonebattery.charge=100.",
1104         "if zonestatus=xxxxlxxxxxxxx,
1105 oic.r.ias.zone.zonepowersource.powerSources=['Internal Battery'] &
1106 oic.r.ias.zone.zonebattery.defect=true & oic.r.ias.zone.zonebattery.charge=100."
1107     ],
1108     "x-from-ocf": [
1109         "N/A"
1110     ]
1111 },
1112 },
1113 "IAS_CIE_address": {
1114     "type": "string",
1115     "description": "Address of IAS Control and Indicating Equipment (CIE)",
1116     "x-ocf-conversion": {
1117         "x-ocf-alias": "oic.r.iaszone",
1118         "x-to-ocf": [
1119             "ocf.iascieaddress= IAS_CIE_address"
1120         ],
1121         "x-from-ocf": [
1122             "N/A"
1123         ]
1124     },
1125 },
1126 "zoneID": {
1127     "type": "integer",
1128     "description": "Unique id allocated by IAS CIE",
1129     "x-ocf-conversion": {
1130         "x-ocf-alias": "oic.r.iaszone",
1131         "x-to-ocf": [
1132             "ocf.zoneid=zoneID"
1133         ],
1134         "x-from-ocf": [
1135             "N/A"
1136         ]
1137     },
1138 },
1139 "numberofzonesensitivitylevelsupported": {
1140     "type": "integer",
1141     "description": "Total number of sensitivity levels supported by the IAS Zone",
1142     "x-ocf-conversion": {
1143         "x-ocf-alias": "oic.r.iaszone",
1144         "x-to-ocf": [
1145             "ocf.numzonesensitivitylevel= numberofzonesensitivitylevelsupported"
1146         ],
1147         "x-from-ocf": [
1148             "N/A"
1149         ]
1150     },
1151 },
1152 "currentzonesensitivitylevel": {
1153     "type": "integer",
1154     "description": "Sensitivity level of IAS Zone",
1155     "x-ocf-conversion": {
1156         "x-ocf-alias": "oic.r.iaszone",
1157         "x-to-ocf": [
1158             "ocf.currentzonesensitivitylevel = currentzonesensitivitylevel"
1159         ],
1160         "x-from-ocf": [
1161             "N/A"
1162         ]
1163     },
1164 },
1165 },
1166 },
1167 },
1168 "type": "object",
1169 "allOf": [
1170     {"$ref": "#/definitions/zcl.iaszone.info"}
1171 ],
1172 "required": [ "zonestate"]
```



1173 }
1174

1175 8.9 Level Control Cluster - Control

1176 8.9.1 Derived model

1177 The derived model: zcl.levelcontrol.control.moveto.

1178 8.9.2 Property definition

1179 **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,t ransitiontime)

1180 **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

1181 8.9.3 Derived model definition

```

1182 {
1183   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.levelcontrol.control.json#",
1184   "$schema": "http://json-schema.org/draft-04/schema#",
1185   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1186   "title": "Level Control Cluster - Control",
1187   "definitions": {
1188     "zcl.levelcontrol.control.moveto": {
1189       "properties": {
1190         "level": {
1191           "type": "integer",
1192           "description": "Move to certain dimming value as fast as possible ",
1193           "x-ocf-conversion": {
1194             "x-ocf-alias": "oic.r.light.dimming",
1195             "x-from-ocf": [
1196               "level=ocf.dimmingSetting * 254 /100 , transitiontime=0",
1197               "zcl.command.levelcontrol::movetolevel(level,transitiontime)"
1198             ],
1199             "x-to-ocf": [
1200               "N/A"
1201             ]
1202           }
1203         }
1204       }
1205     }
1206   },
1207   "type": "object",
1208   "allOf": [
1209     {"$ref": "#/definitions/zcl.levelcontrol.control.movetolevel"}
1210   ]
1211 }
1212

```

1213 8.10 Level Control Cluster - Information

1214 8.10.1 Derived model

1215 The derived model: zcl.levelcontrol.info.



1216 **8.10.2 Property definition**

1217 **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

1218 **Table 22 The properties of zcl.levelcontrol.info.**

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

1219 **8.10.3 Derived model definition**

```

1220 {
1221   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.levelcontrol.info.json#",
1222   "$schema": "http://json-schema.org/draft-04/schema#",
1223   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1224   "title": "Level Control Cluster - Information",
1225   "definitions": {
1226     "zcl.levelcontrol.info": {
1227       "type": "object",
1228       "properties": {
1229         "currentlevel": {
1230           "type": "integer",
1231           "description": "current dimming value",
1232           "x-ocf-conversion": {
1233             "x-ocf-alias": "oic.r.light.dimming",
1234             "x-to-ocf": [
1235               "ocf.dimmingsetting = currentlevel/254 * 100"
1236             ],
1237             "x-from-ocf": [
1238               "N/A"
1239             ]
1240           }
1241         }
1242       }
1243     },
1244   },
1245   "type": "object",
1246   "allOf": [
1247     {"$ref": "#/definitions/zcl.levelcontrol.info"}
1248   ],
1249   "required": [ "currentlevel" ]
1250 }
1251

```

1252 **8.11 Occupancy Sensing Cluster - Information**

1253 **8.11.1 Derived model**

1254 The derived model: zcl.occupancysensing.info.

1255 **8.11.2 Property definition**

1256 **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 =xxxxxx0, then ocf.value = false if occupancy =xxxxxx1, then ocf.value = true	N/A



1257

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

1258

8.11.3 Derived model definition

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270

1271

1272

1273

1274

1275

1276

1277

1278

1279

1280

1281

1282

1283

1284

1285

1286

1287

1288

1289

1290

1291

1292

```
{
  "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.occupancysensing.info.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Occupancy Sensing Cluster - Information",
  "definitions": {
    "zcl.occupancysensing.info": {
      "type": "object",
      "properties": {
        "occupancy": {
          "type": "number",
          "description": "x is a variable. Data type of occupancy in Zigbee is 8 bitmap (xxxxxxx)
while data type of value in OCF is boolean type i.e., true=occupied, false=unoccupied",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.sensor.presence",
            "x-to-ocf": [
              "if occupancy =xxxxxxx0, then ocf.value = false",
              "if occupancy =xxxxxxx1, then ocf.value = true"
            ],
            "x-from-ocf": [
              "N/A"
            ]
          }
        }
      }
    }
  },
  "type": "object",
  "allOf": [
    {"$ref": "#/definitions/zcl.occupancysensing.info"}
  ],
  "required": [ "occupancy" ]
}
```

1293

8.12 On/Off Cluster - Control

1294

8.12.1 Derived model

1295

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

1296

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

1297

8.12.2 Property definition

1298

Table 25 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().



1299

Table 26 The properties of zcl.onoff.control.on.

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

1300

Table 27 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().

1301

Table 28 The properties of zcl.onoff.control.off.

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

1302

8.12.3 Derived model definition

1303

1304

1305

1306

1307

1308

1309

1310

1311

1312

1313

1314

1315

1316

1317

1318

1319

1320

1321

1322

1323

1324

1325

1326

1327

1328

1329

1330

1331

1332

1333

1334

1335

1336

1337

1338

1339

1340

1341

1342

1343

1344

1345

1346

1347

1348

```

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

```



1349 }
1350

1351 8.13 On/off Cluster - Information

1352 8.13.1 Derived model

1353 The derived model: zcl.onoff.

1354 8.13.2 Property definition

1355 **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

1356 **Table 30 The properties of zcl.onoff.**

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

1357 8.13.3 Derived model definition

```

1358 {
1359   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.onoff.info.json#",
1360   "$schema": "http://json-schema.org/draft-04/schema#",
1361   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1362   "title": "On/off Cluster - Information",
1363   "definitions": {
1364     "zcl.onoff": {
1365       "type": "object",
1366       "properties": {
1367         "onoff": {
1368           "type": "boolean",
1369           "description": "On/off status of the device",
1370           "x-ocf-conversion": {
1371             "x-ocf-alias": "oic.r.switch.binary",
1372             "x-to-ocf": [
1373               "if onoff = false, then ocf.value = false",
1374               "if onoff = true, then ocf.value = true"
1375             ],
1376             "x-from-ocf": [
1377               "N/A"
1378             ]
1379           }
1380         }
1381       }
1382     }
1383   },
1384   "type": "object",
1385   "allOf": [
1386     { "$ref": "#/definitions/zcl.onoff.info" }
1387   ],
1388   "required": [ "onoff" ]
1389 }
1390

```

1391 8.14 Temperature Measurement Cluster - Information

1392 8.14.1 Derived model

1393 The derived model: zcl.temperaturemeasurement.info.



1394 **8.14.2 Property definition**

1395 **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
MaxMeasuredValue		oic.r.temperature	ocf.range[1] = MaxMeasuredValue/100	N/A
Tolerance		oic.r.temperature	ocf.precision = Tolerance/100	N/A
MinMeasuredValue		oic.r.temperature	ocf.range[0] = MinMeasuredValue/100	N/A

1396 **Table 32 The properties of zcl.temperaturemeasurement.info.**

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

1397 **8.14.3 Derived model definition**

```

1398 {
1399   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.temperaturemeasurement.info.json#",
1400   "$schema": "http://json-schema.org/draft-04/schema#",
1401   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1402   "title": "Temperature Measurement Cluster - Information",
1403   "definitions": {
1404     "zcl.temperaturemeasurement.info": {
1405       "type": "object",
1406       "properties": {
1407         "MeasuredValue": {
1408           "type": "number",
1409           "description": "Measured value",
1410           "x-ocf-conversion": {
1411             "x-ocf-alias": "oic.r.temperature",
1412             "x-to-ocf": [
1413               "ocf.temperature = MeasuredValue/100",
1414               "units = C"
1415             ],
1416             "x-from-ocf": [
1417               "N/A"
1418             ]
1419           }
1420         },
1421         "Tolerance": {
1422           "type": "number",
1423           "description": "Magnitude of the possible error",
1424           "x-ocf-conversion": {
1425             "x-ocf-alias": "oic.r.temperature",
1426             "x-to-ocf": [
1427               "ocf.precision = Tolerance/100"
1428             ],
1429             "x-from-ocf": [
1430               "N/A"
1431             ]
1432           }
1433         }
1434       }
1435     }
1436   }

```




```

1433     },
1434     "MinMeasuredValue": {
1435       "type": "number",
1436       "description": "Minimum value of MeasuredValue",
1437       "x-ocf-conversion": {
1438         "x-ocf-alias": "oic.r.temperature",
1439         "x-to-ocf": [
1440           "ocf.range[0] = MinMeasuredValue/100"
1441         ],
1442         "x-from-ocf": [
1443           "N/A"
1444         ]
1445       }
1446     },
1447     "MaxMeasuredValue": {
1448       "type": "number",
1449       "description": "Maximum value of MeasuredValue",
1450       "x-ocf-conversion": {
1451         "x-ocf-alias": "oic.r.temperature",
1452         "x-to-ocf": [
1453           "ocf.range[1] = MaxMeasuredValue/100"
1454         ],
1455         "x-from-ocf": [
1456           "N/A"
1457         ]
1458       }
1459     }
1460   }
1461 }
1462 },
1463 "type": "object",
1464 "allOf": [
1465   {"$ref": "#/definitions/zcl.temperaturemeasurement.info"}
1466 ],
1467 "required": [ "MeasuredValue", "Tolerance", "MinMeasuredValue", "MaxMeasuredValue" ]
1468 }
1469

```

1470 8.15 Thermostat Cluster - Cool - Control

1471 8.15.1 Derived model

1472 The derived model: zcl.thermostat_cool.control.setpointraiselower.

1473 8.15.2 Property definition

1474 **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)

1475 **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



1476 8.15.3 Derived model definition

```

1477 {
1478   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.thermostat_cool.control.json#",
1479   "$schema": "http://json-schema.org/draft-04/schema#",
1480   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1481   "title": "Thermostat Cluster - Cool - Control",
1482   "definitions": {
1483     "zcl.thermostat_cool.control.setpointraiselower": {
1484       "type": "object",
1485       "properties": {
1486         "amount": {
1487           "type": "number",
1488           "description": "Set the target temperature with cool mode. Mode=0x01 is set by Zigbee 3.0
1489 translator",
1490           "x-ocf-conversion": {
1491             "x-ocf-alias": "oic.r.temperature",
1492             "x-from-ocf": [
1493               "if ocf.temperature is updated, then amount= ocf.temperature*100.",
1494               "zcl.command.thermostat::setpointraiselower(mode, amount)"
1495             ],
1496             "x-to-ocf": [
1497               "N/A"
1498             ]
1499           }
1500         }
1501       }
1502     }
1503   },
1504   "type": "object",
1505   "allOf": [
1506     {"$ref": "#/definitions/zcl.thermostat_cool.control.setpointraiselower"}
1507   ]
1508 }
1509

```

1510 8.16 Thermostat Cluster - Current Temperature - Information

1511 8.16.1 Derived model

1512 The derived model: zcl.thermostat_currenttemperature.info.

1513 8.16.2 Property definition

1514 **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=localtemperature/100units = C	N/A

1515 **Table 36 The properties of zcl.thermostat_currenttemperature.info.**

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

1516 8.16.3 Derived model definition

```

1517 {
1518   "id":
1519   "http://openinterconnect.org/zigbeemapping/schemas/zcl.thermostat_currenttemperature.info.json#",
1520   "$schema": "http://json-schema.org/draft-04/schema#",
1521   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1522   "title": "Thermostat Cluster - Current Temperature - Information ",
1523   "definitions": {
1524     "zcl.thermostat_currenttemperature.info": {
1525       "type": "object",
1526       "properties": {

```



```

1527     "localtemperature": {
1528       "type": "number",
1529       "description": "current sensed temperature",
1530       "x-ocf-conversion": {
1531         "x-ocf-alias": "oic.r.temperature",
1532         "x-to-ocf": [
1533           "ocf.temperature=localtemperature/100",
1534           "units = C"
1535         ],
1536         "x-from-ocf": [
1537           "N/A"
1538         ]
1539       }
1540     }
1541   }
1542 }
1543 },
1544 "type": "object",
1545 "allOf": [
1546   { "$ref": "#/definitions/zcl.thermostat_currenttemperature.info" }
1547 ],
1548 "required": [ "localtemperature" ]
1549 }
1550

```

1551 8.17 Thermostat Cluster - Heat - Control

1552 8.17.1 Derived model

1553 The derived model: zcl.thermostat_heat.control.setpointraiseLower.

1554 8.17.2 Property definition

1555 **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)

1556 **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

1557 8.17.3 Derived model definition

```

1558 {
1559   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.thermostat_heat.control.json#",
1560   "$schema": "http://json-schema.org/draft-04/schema#",
1561   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1562   "title": "Thermostat Cluster - Heat - Control",
1563   "definitions": {
1564     "zcl.thermostat_heat.control.setpointraiseLower": {
1565       "type": "object",
1566       "properties": {
1567         "amount": {
1568           "type": "number",
1569           "description": "Set the target temperature with heat mode. Mode=0x00 is set by Zigbee 3.0 translator",
1570

```



```

1571     "x-ocf-conversion": {
1572       "x-ocf-alias": "oic.r.temperature",
1573       "x-from-ocf": [
1574         "if ocf.temperature is updated, then amount= ocf.temperature*100.",
1575         "zcl.command.thermostat::setpointraiseLower(mode, amount)"
1576       ],
1577       "x-to-ocf": [
1578         "N/A"
1579       ]
1580     }
1581   }
1582 }
1583 }
1584 },
1585 "type": "object",
1586 "allOf": [
1587   {"$ref": "#/definitions/zcl.thermostat_heat.control.setpointraiseLower"}
1588 ]
1589 }
1590

```

1591 8.18 Window Covering Cluster - Configuration - Control

1592 8.18.1 Derived model

1593 The derived model: zcl.windowcovering_conf.control.

1594 8.18.2 Property definition

1595 **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)
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)
Velocity-Lift	oic.r.windowcovering	N/A	if ocf.liftvelocity is updated, Velocity-Lift = ocf.liftvelocity.zcl.command.general::write(Velocity-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)

1596 **Table 40 The properties of zcl.windowcovering_conf.control.**

Zigbee Property name	Property	Type	Required	Description
Acceleration Time-Lift		integer	no	Set ramp up times to reaching the velocity setting (0.1sec).
Deceleration Time-Lift		integer	no	Set ramp down times associated with



			stopping the velocity setting (0.1sec).
Velocity-Lift	integer	no	Set velocity associated with Lifting the Window Covering (cm/sec).
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.

1597 8.18.3 Derived model definition

```

1598 {
1599   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_conf.control.json#",
1600   "$schema": "http://json-schema.org/draft-04/schema#",
1601   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1602   "title": "Window Covering Cluster - Configuration - Control",
1603   "definitions": {
1604     "zcl.windowcovering_conf.control": {
1605       "properties": {
1606         "mode": {
1607           "type": "integer",
1608           "description": "Set the mode. x is a variable. Data type of Mode in Zigbee is 8 bitmap
1609 (xxxxxxx) while data type of mode in OCF is array with 4 Boolean type items(i.e., [Reversed Motor
1610 Direction, Calibration Mode, Maintenance Mode, LED]). Reversed Motor Direction : 0 = motor direction
1611 is normal, 1 = motor direction is reversed. Calibration Mode : 0 = run in normal mode, 1 = run in
1612 calibration mode. Maintenance Mode : 0 = motor is running normally, 1 = motor is running in
1613 maintenance mode. LED: 0 = LEDs are off, 1 = LEDs will display feedback.",
1614           "x-ocf-conversion": {
1615             "x-ocf-alias": "oic.r.windowcovering",
1616             "x-from-ocf": [
1617               "if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxxxxx0.",
1618               "if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxxxxx1.",
1619               "if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxxxxx0x.",
1620               "if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxxxxx1x.",
1621               "if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxxx0xx.",
1622               "if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxxx1xx.",
1623               "if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxxx0xxx.",
1624               "if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxxx1xxx."

```



```
1625         "zcl.command.general::write(mode)"
1626     ],
1627     "x-to-ocf": [
1628         "N/A"
1629     ]
1630 }
1631 },
1632 "Velocity-Lift": {
1633     "type": "integer",
1634     "description": "Set velocity associated with Lifting the Window Covering (cm/sec).",
1635     "x-ocf-conversion": {
1636         "x-ocf-alias": "oic.r.windowcovering",
1637         "x-from-ocf": [
1638             "if ocf.liftvelocity is updated, Velocity-Lift = ocf.liftvelocity.",
1639             "zcl.command.general::write(Velocity-Lift)"
1640         ],
1641         "x-to-ocf": [
1642             "N/A"
1643         ]
1644     }
1645 },
1646 "Acceleration Time-Lift": {
1647     "type": "integer",
1648     "description": "Set ramp up times to reaching the velocity setting (0.1sec).",
1649     "x-ocf-conversion": {
1650         "x-ocf-alias": "oic.r.windowcovering",
1651         "x-from-ocf": [
1652             "if ocf.liftaccelerationtime is updated, Acceleration Time-
1653 Lift=ocf.liftaccelerationtime.",
1654             "zcl.command.general::write(Acceleration Time-Lift)"
1655         ],
1656         "x-to-ocf": [
1657             "N/A"
1658         ]
1659     }
1660 },
1661 "Deceleration Time-Lift": {
1662     "type": "integer",
1663     "description": "Set ramp down times associated with stoping the velocity setting
1664 (0.1sec).",
1665     "x-ocf-conversion": {
1666         "x-ocf-alias": "oic.r.windowcovering",
1667         "x-from-ocf": [
1668             "if ocf.liftdecelerationtime is updated, Deceleration Time-
1669 Lift=ocf.liftdecelerationtime.",
1670             "zcl.command.general::write(Deceleration Time-Lift)"
1671         ],
1672         "x-to-ocf": [
1673             "N/A"
1674         ]
1675     }
1676 }
1677 }
1678 }
1679 },
1680 "type": "object",
1681 "allOf": [
1682     {"$ref": "#/definitions/zcl.windowcovering_conf.control"}
1683 ]
1684 }
1685
```

1686 8.19 Window Covering Cluster - Configuration - Information

1687 8.19.1 Derived model

1688 The derived model: zcl.windowcovering_conf.info.



1689 **8.19.2 Property definition**

1690 **Table 41 The property mapping for zcl.windowcovering_conf.info.**

Zigbee name	Property	OCF Resource	To OCF	From OCF
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= 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 â€ Lift and Tilt. if WindowCoveringType=0x09, ocf.windowcoveringtype= Projector Screen.	N/A
Acceleration Time-Lift		oic.r.windowcovering	ocf.liftaccelerationtime= Acceleration Time-Lift	N/A
Velocity-Lift		oic.r.windowcovering	ocf.liftvelocity = Velocity-Lift	N/A
Mode		oic.r.windowcovering	if Mode =xxxxxxx0, ocf.mode.motordirection = falseif Mode =xxxxxxx1, 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	N/A
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	N/A



		Config/Status =xxxxx1x, ocf.configstatus.online = trueif Config/Status =xxxxx0xx, ocf.configstatus.rotationdirection = 'normal'if Config/Status =xxxxx1xx, 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 =xxx1xxxx, ocf.configstatus.controllift = 'closedloop'if Config/Status =xx0xxxxx, ocf.configstatus.closedloopliftcontrol = 'timer'if Config/Status =xx1xxxxx, ocf.configstatus.closedloopliftcontrol = 'encoder'if Config/Status =x0xxxxxx, ocf.configstatus.closedlooptiltcontrol = 'timer'if Config/Status =x1xxxxxx, ocf.configstatus.closedlooptiltcontrol = 'encoder'	
Deceleration Time-Lift	oic.r.windowcovering	ocf.liftdecelerationtime=	N/A

1691

Table 42 The properties of zcl.windowcovering_conf.info.

Zigbee name	Property	Type	Required	Description
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 - Lift and Tilt, Projector Screen])
Acceleration Time-Lift		integer	no	Ramp up times to reaching the velocity setting (0.1sec).
Velocity-Lift		integer	no	Velocity associated with Lifting the Window Covering (cm/sec).
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.
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 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. 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. 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. 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. 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.
Deceleration Time-Lift	integer	no	Ramp down times associated with stopping the velocity setting (0.1sec).

1692 8.19.3 Derived model definition

```

1693 {
1694   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_conf.info.json#",
1695   "$schema": "http://json-schema.org/draft-04/schema#",
1696   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1697   "title": "Window Covering Cluster - Configuration - Information",
1698   "definitions": {
1699     "zcl.windowcovering_conf.info": {
1700       "type": "object",
1701       "properties": {
1702         "Windowcoveringtype": {
1703           "type": "string",
1704           "description": "Type of Window Covering(i.e., [Rollershade,RollerShade-2 Motor,
1705 RollerShade-Exterior, RollerShade-Exterior-2 Motor, Drapery, Awning, Shutter, Tilt Blind - Tilt
1706 Only, Tilt Blind â€œ Lift and Tilt, Projector Screen])",
1707           "x-ocf-conversion": {
1708             "x-ocf-alias": "oic.r.windowcovering",
1709             "x-to-ocf": [
1710               "if WindowCoveringType=0x00, ocf.windowcoveringtype= Rollershade.",
1711               "if WindowCoveringType=0x01, ocf.windowcoveringtype= RollerShade-2 Motor.",
1712               "if WindowCoveringType=0x02, ocf.windowcoveringtype= RollerShade-Exterior.",
1713               "if WindowCoveringType=0x03, ocf.windowcoveringtype= RollerShade-Exterior-2 Motor.",
1714               "if WindowCoveringType=0x04, ocf.windowcoveringtype= Drapery.",
1715               "if WindowCoveringType=0x05, ocf.windowcoveringtype= Awning.",
1716               "if WindowCoveringType=0x06, ocf.windowcoveringtype= Shutter.",
1717               "if WindowCoveringType=0x07, ocf.windowcoveringtype= Tilt Blind - Tilt Only.",
1718               "if WindowCoveringType=0x08, ocf.windowcoveringtype= Tilt Blind â€œ Lift and Tilt.",
1719               "if WindowCoveringType=0x09, ocf.windowcoveringtype= Projector Screen."
1720             ],
1721             "x-from-ocf": [
1722               "N/A"
1723             ]
1724           }
1725         }
1726       }
1727     }
1728   }

```



```
1725 },
1726 "Config/Status": {
1727   "type": "integer",
1728   "description": " x is a variable. Config/Status in Zigbee maps to configstatus in OCF.
1729 Data type of Config/Status in Zigbee is 8 bitmap (xxxxxxx) : bit 0 = Operational, bit 1 = Online,
1730 bit 2 = Reversal, bit 3 = Control-Lift, bit 4 = Control-Tilt, bit 5 = Encoder-Lift, bit 6 =
1731 Encoder-Tilt. Operational: This status bit defines if the Window Covering is operational. 0 = Not
1732 Operational, 1 = Operational. Online: This status bit defines if the Window Covering is enabled for
1733 transmitting over the ZigBee network. 0 = Not Online, 1 = Online. Reversal: This status bit
1734 identifies if the direction of rotation for the Window Covering has been reversed in order for
1735 Open/Up commands to match the physical installation condition. 0 = Commands are normal, 1 = Open/Up
1736 Commands have been reversed. Control Lift: This status bit identifies if the window covering
1737 supports Open Loop or Closed Loop Lift Control. 0 = Lift control is Open Loop, 1 = Lift control is
1738 Closed. Control Tilt: This status bit identifies if the window covering supports Open Loop or Closed
1739 Loop Tilt Control. 0 = Tilt control is Open Loop, 1 = Tilt control is Closed. Encoder Lift: This
1740 status bit identifies if a Closed Loop Controlled Window Covering is employing an encoder for
1741 positioning the height of the window covering. 0 = Timer Controlled, 1 = Encoder Controlled. Encoder
1742 Tilt: This status bit identifies if a Closed Loop Controlled Window Covering is employing an encoder
1743 for tilting the window covering. 0 = Timer Controlled, 1 = Encoder Controlled.",
1744   "x-ocf-conversion": {
1745     "x-ocf-alias": "oic.r.windowcovering",
1746     "x-to-ocf": [
1747       "if Config/Status =xxxxxxx0, ocf.configstatus.operational = false",
1748       "if Config/Status =xxxxxxx1, ocf.configstatus.operational = true",
1749       "if Config/Status =xxxxxxx0x, ocf.configstatus.online = false",
1750       "if Config/Status =xxxxxxx1x, ocf.configstatus.online = true",
1751       "if Config/Status =xxxxxx0xx, ocf.configstatus.rotationdirection = 'normal'",
1752       "if Config/Status =xxxxxx1xx, ocf.configstatus.rotationdirection = 'reversed'",
1753       "if Config/Status =xxxxx0xxx, ocf.configstatus.controllift = 'openloop'",
1754       "if Config/Status =xxxxx1xxx, ocf.configstatus.controllift = 'closedloop'",
1755       "if Config/Status =xxx0xxxx, ocf.configstatus.controltilt = 'openloop'",
1756       "if Config/Status =xxx1xxxx, ocf.configstatus.controltilt = 'closedloop'",
1757       "if Config/Status =xx0xxxxx, ocf.configstatus.closedloopliftcontrol = 'timer'",
1758       "if Config/Status =xx1xxxxx, ocf.configstatus.closedloopliftcontrol = 'encoder'",
1759       "if Config/Status =x0xxxxxx, ocf.configstatus.closedlooptiltcontrol = 'timer'",
1760       "if Config/Status =x1xxxxxx, ocf.configstatus.closedlooptiltcontrol = 'encoder'"
1761     ],
1762     "x-from-ocf": [
1763       "N/A"
1764     ]
1765   }
1766 },
1767 "Mode": {
1768   "type": "integer",
1769   "description": " x is a variable. Mode in Zigbee maps to mode in OCF. Data type of Mode in
1770 Zigbee is 8 bitmap (xxxxxxx) : bit 0 = Reversed Motor Direction, bit 1 = Calibration Mode, bit 2 =
1771 Maintenance Mode, bit 3 = LED. Reversed Motor Direction : 0 = motor direction is normal, 1 = motor
1772 direction is reversed. Calibration Mode : 0 = run in normal mode, 1 = run in calibration mode.
1773 Maintenance Mode : 0 = motor is running normally, 1 = motor is running in maintenance mode. LED: 0 =
1774 LEDs are off, 1 = LEDs will display feedback.",
1775   "x-ocf-conversion": {
1776     "x-ocf-alias": "oic.r.windowcovering",
1777     "x-to-ocf": [
1778       "if Mode =xxxxxxx0, ocf.mode.motordirection = false",
1779       "if Mode =xxxxxxx1, ocf.mode.motordirection = true",
1780       "if Mode =xxxxxxx0x, ocf.mode.calibration = false",
1781       "if Mode =xxxxxxx1x, ocf.mode.calibration = true",
1782       "if Mode =xxxxx0xx, ocf.mode.maintenance = false",
1783       "if Mode =xxxxx1xx, ocf.mode.maintenance = true",
1784       "if Mode =xxxx0xxx, ocf.mode.ledfeedback = false",
1785       "if Mode =xxxx1xxx, ocf.mode.ledfeedback = true"
1786     ],
1787     "x-from-ocf": [
1788       "N/A"
1789     ]
1790   }
1791 },
1792 "Velocity-Lift": {
1793   "type": "integer",
1794   "description": "Velocity associated with Lifting the Window Covering (cm/sec).",
1795   "x-ocf-conversion": {
```



```

1796         "x-ocf-alias": "oic.r.windowcovering",
1797         "x-to-ocf": [
1798             "ocf.liftvelocity = Velocity-Lift"
1799         ],
1800         "x-from-ocf": [
1801             "N/A"
1802         ]
1803     },
1804 },
1805 "Acceleration Time-Lift": {
1806     "type": "integer",
1807     "description": "Ramp up times to reaching the velocity setting (0.1sec).",
1808     "x-ocf-conversion": {
1809         "x-ocf-alias": "oic.r.windowcovering",
1810         "x-to-ocf": [
1811             "ocf.liftaccelerationtime= Acceleration Time-Lift"
1812         ],
1813         "x-from-ocf": [
1814             "N/A"
1815         ]
1816     },
1817 },
1818 "Deceleration Time-Lift": {
1819     "type": "integer",
1820     "description": "Ramp down times associated with stoping the velocity setting (0.1sec).",
1821     "x-ocf-conversion": {
1822         "x-ocf-alias": "oic.r.windowcovering",
1823         "x-to-ocf": [
1824             "ocf.liftdecelerationtime= Deceleration Time-Lift"
1825         ],
1826         "x-from-ocf": [
1827             "N/A"
1828         ]
1829     },
1830 },
1831 }
1832 }
1833 },
1834 "type": "object",
1835 "allOf": [
1836     {"$ref": "#/definitions/zcl.windowcovering_conf.info"}
1837 ],
1838 "required": [ "Windowcoveringtype", "Config/Status", "Mode" ]
1839 }
1840

```

1841 **8.20 Window Covering Cluster - Lift Percentage - Control**

1842 **8.20.1 Derived model**

1843 The derived model: /zcl.windowcovering_liftpercentage.control.gotoliftpercentage.

1844 **8.20.2 Property definition**

1845 **Table 43 The property mapping for**
1846 **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)

1847 **Table 44 The properties of zcl.windowcovering_liftpercentage.control.gotoliftpercentage.**

Zigbee name	Property	Type	Required	Description
-------------	----------	------	----------	-------------



percentageliftvalue	integer	no	Adjust the window at the percentage lift value.
---------------------	---------	----	---

1848 **8.20.3 Derived model definition**

```

1849 {
1850   "id":
1851   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_liftpercentage.control.json#",
1852   "$schema": "http://json-schema.org/draft-04/schema#",
1853   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1854   "title": "Window Covering Cluster - Lift Percentage - Control",
1855   "definitions": {
1856     "/zcl.windowcovering_liftpercentage.control.gotoliftpercentage": {
1857       "properties": {
1858         "percentageliftvalue": {
1859           "type": "integer",
1860           "description": "Adjust the window at the percentage lift value.",
1861           "x-ocf-conversion": {
1862             "x-ocf-alias": "oic.r.openlevel",
1863             "x-from-ocf": [
1864               "if ocf.openLevel is updated, percentage lift value = ocf.openLevel.",
1865               "zcl.command.windowcovering::gotoliftpercentage(percentageliftvalue)"
1866             ],
1867             "x-to-ocf": [
1868               "N/A"
1869             ]
1870           }
1871         }
1872       }
1873     },
1874     "type": "object",
1875     "allOf": [
1876       {"$ref": "#/definitions/zcl.windowcovering_liftpercentage.control.gotoliftpercentage"}
1877     ]
1878   }
1879 }
1880

```

1881 **8.21 Window Covering Cluster - Lift Percentage - Information**

1882 **8.21.1 Derived model**

1883 The derived model: zcl.windowcovering_liftpercentage.info.

1884 **8.21.2 Property definition**

1885 **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

1886 **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

1887 **8.21.3 Derived model definition**

```

1888 {
1889   "id":
1890   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_liftpercentage.info.json#",
1891   "$schema": "http://json-schema.org/draft-04/schema#",

```



```

1892 "description" : "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1893 "title": "Window Covering Cluster - Lift Percentage - Information",
1894 "definitions": {
1895   "zcl.windowcovering_liftpercentage.info": {
1896     "type": "object",
1897     "properties": {
1898       "CurrentPositionLiftPercentage": {
1899         "type": "integer",
1900         "description": "Position as a percentage between InstalledOpenLimit-Lift and
1901 InstalledClosedLimit-Lift",
1902         "x-ocf-conversion": {
1903           "x-ocf-alias": "oic.r.openlevel",
1904           "x-to-ocf": [
1905             "ocf.openLevel= CurrentPositionLiftPercentage"
1906           ],
1907           "x-from-ocf": [
1908             "N/A"
1909           ]
1910         }
1911       }
1912     }
1913   },
1914   "type": "object",
1915   "allOf": [
1916     { "$ref": "#/definitions/zcl.windowcovering_liftpercentage.info" }
1917   ],
1918   "required": ["CurrentPositionLiftPercentage"]
1919 }
1920 }
1921

```

1922 8.22 Window Covering Cluster - Lift Position - Control

1923 8.22.1 Derived model

1924 The derived model: zcl.windowcovering_liftposition.control.gotoliftvalue.

1925 8.22.2 Property definition

1926 **Table 47** The property mapping for 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(liftvalue)

1927 **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.

1928 8.22.3 Derived model definition

```

1929 {
1930   "id":
1931   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_liftposition.control.json#",
1932   "$schema": "http://json-schema.org/draft-04/schema#",
1933   "description" : "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1934   "title": "Window Covering Cluster - Lift Position - Control",
1935   "definitions": {
1936     "zcl.windowcovering_liftposition.control.gotoliftvalue": {
1937       "properties": {
1938         "liftvalue": {
1939           "type": "integer",
1940           "description": "Adjust the window at the lift value.",

```



```

1941     "x-ocf-conversion": {
1942       "x-ocf-alias": "oic.r.openlevel",
1943       "x-from-ocf": [
1944         "if ocf.openLevel is updated, lift value= ocf.openLevel.",
1945         "zcl.command.windowcovering::gotoliftvalue(liftvalue)"
1946       ],
1947       "x-to-ocf": [
1948         "N/A"
1949       ]
1950     }
1951   }
1952 }
1953 }
1954 },
1955 "type": "object",
1956 "allOf": [
1957   {"$ref": "#/definitions/zcl.windowcovering_liftposition.control.gotoliftvalue"}
1958 ]
1959 }
1960

```

1961 8.23 Window Covering Cluster - Lift Position - Information

1962 8.23.1 Derived model

1963 The derived model: /zcl.windowcovering_liftposition.info.

1964 8.23.2 Property definition

1965 **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
InstalledOpenLimit-Lift		oic.r.openlevel	ocf.range[1]= InstalledOpenLimit-Lift	N/A
InstalledClosedLimit-Lift		oic.r.openlevel	ocf.range[0]= InstalledClosedLimit-Lift	N/A

1966 **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)
InstalledOpenLimit-Lift		integer	yes	Open limit for lifting the Window Covering (cm)
InstalledClosedLimit-Lift		integer	yes	Close limit for lifting the Window Covering (cm)

1967 8.23.3 Derived model definition

```

1968 {
1969   "id":
1970   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_liftposition.info.json#",
1971   "$schema": "http://json-schema.org/draft-04/schema#",
1972   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1973   "title": "Window Covering Cluster - Lift Position - Information",
1974   "definitions": {
1975     "/zcl.windowcovering_liftposition.info": {

```



```

1976     "type": "object",
1977     "properties": {
1978       "InstalledClosedLimit-Lift": {
1979         "type": "integer",
1980         "description": "Close limit for lifting the Window Covering (cm)",
1981         "x-ocf-conversion": {
1982           "x-ocf-alias": "oic.r.openlevel",
1983           "x-to-ocf": [
1984             "ocf.range[0]= InstalledClosedLimit-Lift"
1985           ],
1986           "x-from-ocf": [
1987             "N/A"
1988           ]
1989         }
1990       },
1991       "InstalledOpenLimit-Lift": {
1992         "type": "integer",
1993         "description": "Open limit for lifting the Window Covering (cm)",
1994         "x-ocf-conversion": {
1995           "x-ocf-alias": "oic.r.openlevel",
1996           "x-to-ocf": [
1997             "ocf.range[1]= InstalledOpenLimit-Lift"
1998           ],
1999           "x-from-ocf": [
2000             "N/A"
2001           ]
2002         }
2003       },
2004       "CurrentPosition-Lift": {
2005         "type": "integer",
2006         "description": "Position of Window Covering from the top of the shade (cm)",
2007         "x-ocf-conversion": {
2008           "x-ocf-alias": "oic.r.openlevel",
2009           "x-to-ocf": [
2010             "ocf.openLevel= CurrentPosition-Lift"
2011           ],
2012           "x-from-ocf": [
2013             "N/A"
2014           ]
2015         }
2016       }
2017     }
2018   },
2019   "type": "object",
2020   "allOf": [
2021     {"$ref": "#/definitions/zcl.windowcovering_liftposition.info"}
2022   ],
2023   "required": [ "InstalledClosedLimit-Lift", "InstalledOpenLimit-Lift", "CurrentPosition-Lift" ]
2024 }
2025 }
2026

```

2027 8.24 Window Covering Cluster - Tilt Percentage - Control

2028 8.24.1 Derived model

2029 The derived model: zcl.windowcovering_tiltpercentage.control.gototiltpercentage.

2030 8.24.2 Property definition

2031 **Table 51 The property mapping for**
 2032 **zcl.windowcovering_tiltpercentage.control.gototiltpercentage.**

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



2033 **Table 52 The properties of zcl.windowcovering_tiltpercentage.control.gototiltpercentage.**

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

2034 **8.24.3 Derived model definition**

```

2035 {
2036   "id":
2037   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_tiltpercentage.control.json#",
2038   "$schema": "http://json-schema.org/draft-04/schema#",
2039   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2040   "title": "Window Covering Cluster - Tilt Percentage - Control",
2041   "definitions": {
2042     "zcl.windowcovering_tiltpercentage.control.gototiltpercentage": {
2043       "properties": {
2044         "percentagetiltvalue": {
2045           "type": "integer",
2046           "description": "Adjust the window at the percentage tilt value.",
2047           "x-ocf-conversion": {
2048             "x-ocf-alias": "oic.r.openlevel",
2049             "x-from-ocf": [
2050               "if ocf.openLevel is updated, percentage tilt value = ocf.openLevel.",
2051               "zcl.command.windowcovering::gototiltpercentage(percentagetiltvalue)"
2052             ],
2053             "x-to-ocf": [
2054               "N/A"
2055             ]
2056           }
2057         }
2058       }
2059     },
2060   },
2061   "type": "object",
2062   "allOf": [
2063     {"$ref": "#/definitions/zcl.windowcovering_tiltpercentage.control.gototiltpercentage"}
2064   ]
2065 }
2066

```

2067 **8.25 Window Covering Cluster - Tilt Percentage - Information**

2068 **8.25.1 Derived model**

2069 The derived model: zcl.windowcovering_tiltpercentage.info.

2070 **8.25.2 Property definition**

2071 **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

2072 **Table 54 The properties of zcl.windowcovering_tiltpercentage.info.**

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

2073 **8.25.3 Derived model definition**

```

2074 {
2075   "id":

```



```

2076 "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_tiltpercentage.info.json#",
2077 "$schema": "http://json-schema.org/draft-04/schema#",
2078 "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2079 "title": "Window Covering Cluster - Tilt Percentage - Information",
2080 "definitions": {
2081   "zcl.windowcovering_tiltpercentage.info": {
2082     "type": "object",
2083     "properties": {
2084       "CurrentPositionTiltPercentage": {
2085         "type": "integer",
2086         "description": "Tilt position as a percentage",
2087         "x-ocf-conversion": {
2088           "x-ocf-alias": "oic.r.openlevel",
2089           "x-to-ocf": [
2090             "ocf.openlevel=CurrentPositionTiltPercentage"
2091           ],
2092           "x-from-ocf": [
2093             "N/A"
2094           ]
2095         }
2096       }
2097     }
2098   },
2099   "type": "object",
2100   "allOf": [
2101     { "$ref": "#/definitions/zcl.windowcovering_tiltpercentage.info" }
2102   ],
2103   "required": ["CurrentPositionTiltPercentage"]
2104 }
2105
2106

```

2107 8.26 Window Covering Cluster - Tilt Position - Control

2108 8.26.1 Derived model

2109 The derived model: zcl.windowcovering_tiltposition.control.gototiltvalue.

2110 8.26.2 Property definition

2111 **Table 55** The property mapping for 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)

2112 **Table 56** The properties of zcl.windowcovering_tiltposition.control.gototiltvalue.

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

2113 8.26.3 Derived model definition

```

2114 {
2115   "id":
2116   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_tiltposition.control.json#",
2117   "$schema": "http://json-schema.org/draft-04/schema#",
2118   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2119   "title": "Window Covering Cluster - Tilt Position - Control",
2120   "definitions": {
2121     "zcl.windowcovering_tiltposition.control.gototiltvalue": {
2122       "properties": {
2123         "tiltvalue": {
2124           "type": "integer",

```



```

2125     "description": "Adjust the window at the tilt value.",
2126     "x-ocf-conversion": {
2127       "x-ocf-alias": "oic.r.openlevel",
2128       "x-from-ocf": [
2129         "if ocf.openLevel is updated, tiltvalue= ocf.openLevel.",
2130         "zb.command.windowcovering::gototiltvalue(tiltvalue)"
2131       ],
2132       "x-to-ocf": [
2133         "N/A"
2134       ]
2135     }
2136   }
2137 }
2138 }
2139 },
2140 "type": "object",
2141 "allOf": [
2142   {"$ref": "#/definitions/zcl.windowcovering_tiltposition.control.gototiltvalue"}
2143 ]
2144 }
2145

```

2146 8.27 Window Covering Cluster - Tilt Position - Information

2147 8.27.1 Derived model

2148 The derived model: zcl.windowcovering_tiltposition.info.

2149 8.27.2 Property definition

2150 **Table 57 The property mapping for zcl.windowcovering_tiltposition.info.**

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

2151 **Table 58 The properties of zcl.windowcovering_tiltposition.info.**

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

2152 8.27.3 Derived model definition

```

2153 {
2154   "id":
2155   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_tiltposition.info.json#",
2156   "$schema": "http://json-schema.org/draft-04/schema#",
2157   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2158   "title": "Window Covering Cluster - Tilt Position - Information",
2159   "definitions": {
2160     "zcl.windowcovering_tiltposition.info": {
2161       "type": "object",
2162       "properties": {
2163         "InstalledOpenLimit-Tilt": {
2164           "type": "integer",
2165           "description": "Close limit for tilting the Window Covering (0.1 degree)",
2166           "x-ocf-conversion": {

```



```
2167         "x-ocf-alias": "oic.r.openlevel",
2168         "x-to-ocf": [
2169             "ocf.range[0] = InstalledClosedLimit-Tilt"
2170         ],
2171         "x-from-ocf": [
2172             "N/A"
2173         ]
2174     },
2175 },
2176 "InstalledOpenLimit-Tilt": {
2177     "type": "integer",
2178     "description": "Open limit for tilting the Window Covering (0.1 degree)",
2179     "x-ocf-conversion": {
2180         "x-ocf-alias": "oic.r.openlevel",
2181         "x-to-ocf": [
2182             "ocf.range[1]= InstalledOpenLimit-Tilt"
2183         ],
2184         "x-from-ocf": [
2185             "N/A"
2186         ]
2187     }
2188 },
2189 "CurrentPosition-Tilt ": {
2190     "type": "integer",
2191     "description": "Tilt position of Window Covering from open (0.1 degree)",
2192     "x-ocf-conversion": {
2193         "x-ocf-alias": "oic.r.openlevel",
2194         "x-to-ocf": [
2195             "ocf.openlevel= CurrentPosition-Tilt"
2196         ],
2197         "x-from-ocf": [
2198             "N/A"
2199         ]
2200     }
2201 }
2202 }
2203 }
2204 },
2205 "type": "object",
2206 "allof": [
2207     {"$ref": "#/definitions/zcl.windowcovering_tiltposition.info"}
2208 ],
2209 "required": [ "InstalledClosedLimit-Tilt", "InstalledOpenLimit-Tilt", "CurrentPosition-Tilt" ]
2210 }
2211
2212
```