

OCF Resource to Zigbee Cluster Mapping Specification

VERSION 2.1.1 | February 2020



OPEN CONNECTIVITY
FOUNDATION™

CONTACT admin@openconnectivity.org

Copyright Open Connectivity Foundation, Inc. © 2020.
All Rights Reserved.

3 Legal Disclaimer

4

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

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

18 Copyright © 2019-2020 Open Connectivity Foundation, Inc. All rights reserved.

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

20

CONTENTS

22	1	Scope	1
23	2	Normative references	1
24	3	Terms, definitions symbols and abbreviations	1
25	3.1	Terms and definitions	1
26	4	Document conventions and organization	2
27	4.1	Conventions	2
28	4.2	Notation.....	2
29	5	Theory of Operation	3
30	5.1	Interworking Approach.....	3
31	5.2	Mapping Syntax.....	3
32	5.2.1	Introduction	3
33	5.2.2	General.....	4
34	5.2.3	Value Assignment	4
35	5.2.4	Property Naming	4
36	5.2.5	Range	4
37	5.2.6	Arrays	4
38	5.2.7	Default Mapping	4
39	5.2.8	Conditional Mapping.....	4
40	5.2.9	Method Invocation.....	4
41	6	Zigbee Translation	4
42	6.1	Operational Scenarios	4
43	6.2	Requirements specific to Zigbee Bridging Function.....	5
44	6.2.1	Requirements specific to Zigbee	5
45	6.2.2	Exposing Zigbee 3.0 Servers to OCF Clients	5
46	6.2.3	Translation for well-defined set	7
47	6.2.4	Exposing a Zigbee 3.0 Server as a Virtual OCF Server	8
48	7	Device Type Mapping.....	14
49	7.1	Introduction	14
50	7.2	Zigbee Device Types to OCF Device Types.....	14
51	8	Resource to ZigBee Cluster Equivalence.....	15
52	8.1	Introduction	15
53	8.2	Zigbee Clusters to OCF Resources.....	15
54	8.2.1	Introduction	15
55	8.2.2	On/off.....	16
56	8.2.3	Level Control.....	16
57	8.2.4	Color Control.....	17
58	8.2.5	Thermostat.....	17
59	8.2.6	Window Covering	18
60	8.2.7	Temperature Measurement	18
61	8.2.8	Occupancy Sensing.....	19
62	8.2.9	IAS Zone.....	19
63	9	Detailed Mapping APIs.....	19

64	9.1	below.....	19
65	9.2	Introduction	19
66	9.3	Color Control Cluster - Color Space - Control	19
67	9.3.1	Derived model	19
68	9.3.2	Property definition	19
69	9.3.3	Derived model definition.....	20
70	9.4	Color Control Cluster - Color Space - Information	21
71	9.4.1	Derived model	21
72	9.4.2	Property definition	21
73	9.4.3	Derived model definition.....	21
74	9.5	Color Control Cluster - Color Temperature - Information	22
75	9.5.1	Derived model	22
76	9.5.2	Property definition	22
77	9.5.3	Derived model definition.....	22
78	9.6	Color Control Cluster - Color Temperature - Information	23
79	9.6.1	Derived model	23
80	9.6.2	Property definition	23
81	9.6.3	Derived model definition.....	24
82	9.7	Color Control Cluster - Hue and Saturation - Control	24
83	9.7.1	Derived model	24
84	9.7.2	Property definition	24
85	9.7.3	Derived model definition.....	25
86	9.8	Color Control Cluster - Hue and Saturation - Information	26
87	9.8.1	Derived model	26
88	9.8.2	Property definition	26
89	9.8.3	Derived model definition.....	26
90	9.9	IAS Zone Cluster - Control.....	27
91	9.9.1	Derived model	27
92	9.9.2	Property definition	27
93	9.9.3	Derived model definition.....	27
94	9.10	IAS Zone Cluster - Information	28
95	9.10.1	Derived model	28
96	9.10.2	Property definition	28
97	9.10.3	Derived model definition.....	34
98	9.11	Level Control Cluster - Control	38
99	9.11.1	Derived model	38
100	9.11.2	Property definition	38
101	9.11.3	Derived model definition.....	38
102	9.12	Level Control Cluster - Information	39
103	9.12.1	Derived model	39
104	9.12.2	Property definition	39
105	9.12.3	Derived model definition.....	39
106	9.13	Occupancy Sensing Cluster - Information	40
107	9.13.1	Derived model	40

108	9.13.2	Property definition	40
109	9.13.3	Derived model definition	40
110	9.14	On/Off Cluster - Control	41
111	9.14.1	Derived model	41
112	9.14.2	Property definition	41
113	9.14.3	Derived model definition	41
114	9.15	On/off Cluster - Information	42
115	9.15.1	Derived model	42
116	9.15.2	Property definition	42
117	9.15.3	Derived model definition	42
118	9.16	Temperature Measurement Cluster - Information	43
119	9.16.1	Derived model	43
120	9.16.2	Property definition	43
121	9.16.3	Derived model definition	44
122	9.17	Thermostat Cluster - Cool - Control	45
123	9.17.1	Derived model	45
124	9.17.2	Property definition	45
125	9.17.3	Derived model definition	45
126	9.18	Thermostat Cluster - Current Temperature - Information	46
127	9.18.1	Derived model	46
128	9.18.2	Property definition	46
129	9.18.3	Derived model definition	46
130	9.19	Thermostat Cluster - Heat - Control	47
131	9.19.1	Derived model	47
132	9.19.2	Property definition	47
133	9.19.3	Derived model definition	47
134	9.20	Window Covering Cluster - Configuration - Control	48
135	9.20.1	Derived model	48
136	9.20.2	Property definition	48
137	9.20.3	Derived model definition	49
138	9.21	Window Covering Cluster - Configuration - Information	50
139	9.21.1	Derived model	50
140	9.21.2	Property definition	50
141	9.21.3	Derived model definition	54
142	9.22	Window Covering Cluster - Lift Percentage - Control	56
143	9.22.1	Derived model	56
144	9.22.2	Property definition	56
145	9.22.3	Derived model definition	57
146	9.23	Window Covering Cluster - Lift Percentage - Information	57
147	9.23.1	Derived model	57
148	9.23.2	Property definition	57
149	9.23.3	Derived model definition	58
150	9.24	Window Covering Cluster - Lift Position - Control	58
151	9.24.1	Derived model	58

152	9.24.2	Property definition	58
153	9.24.3	Derived model definition	59
154	9.25	Window Covering Cluster - Lift Position - Information	59
155	9.25.1	Derived model	59
156	9.25.2	Property definition	59
157	9.25.3	Derived model definition	60
158	9.26	Window Covering Cluster - Tilt Percentage - Control	61
159	9.26.1	Derived model	61
160	9.26.2	Property definition	61
161	9.26.3	Derived model definition	61
162	9.27	Window Covering Cluster - Tilt Percentage - Information	62
163	9.27.1	Derived model	62
164	9.27.2	Property definition	62
165	9.27.3	Derived model definition	62
166	9.28	Window Covering Cluster - Tilt Position - Control	63
167	9.28.1	Derived model	63
168	9.28.2	Property definition	63
169	9.28.3	Derived model definition	63
170	9.29	Window Covering Cluster - Tilt Position - Information	64
171	9.29.1	Derived model	64
172	9.29.2	Property definition	64
173	9.29.3	Derived model definition	64
174			
175			

176

Figures

177

No table of figures entries found.

Tables

179	Table 1 – Translation Rule between Zigbee and OCF Data Models.....	6
180	Table 2 – Zigbee to OCF Mapping Example (Color Temperature Light).....	6
181	Table 3 – Zigbee 3.0 Device & Cluster – OCF Device & Resource mapping.....	7
182	Table 4 – "oic.wk.p" Resource Type mapping	9
183	Table 5 – "oic.wk.d" Resource Type mapping	10
184	Table 6 – "oic.wk.con" Resource Type mapping.....	13
185	Table 7 – Zigbee to OCF Device Type Mapping.....	14
186	Table 8 – Zigbee Server Cluster to OCF Resource Type Mapping	16
187	Table 9 – The Property mapping for "zcl.colorcontrol_csc.control.movetocolor".	19
188	Table 10 – The Properties of "zcl.colorcontrol_csc.control.movetocolor".....	20
189	Table 11 – The Property mapping for "zcl.colorcontrol_csc.info".	21
190	Table 12 – The Properties of "zcl.colorcontrol_csc.info".	21
191	Table 13 – The Property mapping for	
192	"zcl.colorcontrol_ct.control.movetocolortemperature".....	22
193	Table 14 – The Properties of "zcl.colorcontrol_ct.control.movetocolortemperature".	22
194	Table 15 – The Property mapping for "zcl.colorcontrol_ct.info".	23
195	Table 16 – The Properties of "zcl.colorcontrol_ct.info".....	23
196	Table 17 – The Property mapping for	
197	"zcl.colorcontrol_hs.control.movetohueandsaturation".	25
198	Table 18 – The Properties of "zcl.colorcontrol_hs.control.movetohueandsaturation".	25
199	Table 19 – The Property mapping for "zcl.colorcontrol_hs.info".	26
200	Table 20 – The Properties of "zcl.colorcontrol_hs.info".	26
201	Table 21 – The Property mapping for "zcl.iaszone.control".	27
202	Table 22 – The Properties of "zcl.iaszone.control".	27
203	Table 23 – The Property mapping for "zcl.iaszone.info".	28
204	Table 24 – The Properties of "zcl.iaszone.info".....	33
205	Table 25 – The Property mapping for "zcl.levelcontrol.control.moveto".	38
206	Table 26 – The Properties of "zcl.levelcontrol.control.moveto".....	38
207	Table 27 – The Property mapping for "zcl.levelcontrol.info".	39
208	Table 28 – The Properties of "zcl.levelcontrol.info".	39
209	Table 29 – The Property mapping for "zcl.occupancysensing.info".....	40
210	Table 30 – The Properties of "zcl.occupancysensing.info".	40
211	Table 31 – The Property mapping for "zcl.onoff.control.off".	41
212	Table 32 – The Properties of "zcl.onoff.control.off".	41
213	Table 33 – The Property mapping for "zcl.onoff.control.on".....	41
214	Table 34 – The Properties of "zcl.onoff.control.on".	41
215	Table 35 – The Property mapping for "zcl.onoff".	42
216	Table 36 – The Properties of "zcl.onoff".....	42

217	Table 37 – The Property mapping for "zcl.temperaturemeasurement.info".	43
218	Table 38 – The Properties of "zcl.temperaturemeasurement.info".	43
219	Table 39 – The Property mapping for "zcl.thermostat_cool.control.setpointraiselower".	45
220	Table 40 – The Properties of "zcl.thermostat_cool.control.setpointraiselower".	45
221	Table 41 – The Property mapping for "zcl.thermostat_currenttemperature.info".	46
222	Table 42 – The Properties of "zcl.thermostat_currenttemperature.info".	46
223	Table 43 – The Property mapping for "zcl.thermostat_heat.control.setpointraiseLower".	47
224	Table 44 – The Properties of "zcl.thermostat_heat.control.setpointraiseLower".	47
225	Table 45 – The Property mapping for "zcl.windowcovering_conf.control".	48
226	Table 46 – The Properties of "zcl.windowcovering_conf.control".	48
227	Table 47 – The Property mapping for "zcl.windowcovering_conf.info".	51
228	Table 48 – The Properties of "zcl.windowcovering_conf.info".	52
229	Table 49 – The Property mapping for	
230	"zcl.windowcovering_liftpercentage.control.gotoliftpercentage".	56
231	Table 50 – The Properties of	
232	"zcl.windowcovering_liftpercentage.control.gotoliftpercentage".	57
233	Table 51 – The Property mapping for "zcl.windowcovering_liftpercentage.info".	57
234	Table 52 – The Properties of "zcl.windowcovering_liftpercentage.info".	58
235	Table 53 – The Property mapping for	
236	"zcl.windowcovering_liftposition.control.gotoliftvalue".	58
237	Table 54 – The Properties of "zcl.windowcovering_liftposition.control.gotoliftvalue".	59
238	Table 55 – The Property mapping for "zcl.windowcovering_liftposition.info".	59
239	Table 56 – The Properties of "zcl.windowcovering_liftposition.info".	60
240	Table 57 – The Property mapping for	
241	"zcl.windowcovering_tiltpercentage.control.gototiltpercentage".	61
242	Table 58 – The Properties of	
243	"zcl.windowcovering_tiltpercentage.control.gototiltpercentage".	61
244	Table 59 – The Property mapping for "zcl.windowcovering_tiltpercentage.info".	62
245	Table 60 – The Properties of "zcl.windowcovering_tiltpercentage.info".	62
246	Table 61 – The Property mapping for	
247	"zcl.windowcovering_tiltposition.control.gototiltvalue".	63
248	Table 62 – The Properties of "zcl.windowcovering_tiltposition.control.gototiltvalue".	63
249	Table 63 – The Property mapping for "zcl.windowcovering_tiltposition.info".	64
250	Table 64 – The Properties of "zcl.windowcovering_tiltposition.info".	64
251		
252		

253 **1 Scope**

254 This document provides detailed mapping information between Zigbee defined Clusters and OCF
255 defined Resources.

256 **2 Normative references**

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

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

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

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

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

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

282 Derived Models for Interoperability between IoT Ecosystems, Stevens & Merriam, March 2016
283 [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)
284 [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)

285 Zigbee, *Zigbee Specification*, August 2015
286 <http://www.zigbee.org/zigbee-for-developers/zigbee-3-0/>

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

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

291 **3 Terms, definitions symbols and abbreviations**

292 **3.1 Terms and definitions**

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

295 ISO and IEC maintain terminological databases for use in standardization at the following
296 addresses:

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

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

299

300 **3.1.1 Zigbee Attribute**

301 data entity which represents a physical quantity or state within Zigbee. This data is communicated
302 to other devices using commands.

303 **3.1.2 Zigbee Cluster**

304 one or more Zigbee Attributes, commands, behaviours, and dependencies, which supports an
305 independent utility or application function. The term may also be used for an implementation or
306 instance of such on an endpoint.

307 **3.1.3 Zigbee Server**

308 cluster interface which is listed in the input cluster list of the simple descriptor on an endpoint.
309 Typically this interface supports all or most of the attributes of the cluster. A server cluster
310 communicates with a corresponding remote client cluster with the same identifier.

311 **3.1.4 Zigbee 3.0 Server**

312 Zigbee Server which is built on Zigbee 3.0 stack

313 **3.1.5 Zigbee Client**

314 cluster interface which is listed in the output cluster list of the simple descriptor on an endpoint.
315 Typically this interface sends commands that manipulate the attributes on the corresponding
316 Zigbee Server. A client cluster communicates with a corresponding remote server cluster with the
317 same identifier.

318 **3.1.6 Zigbee 3.0 Client**

319 Zigbee Client which is built on Zigbee 3.0 stack

320 **3.1.7 Zigbee Device**

321 unique device identifier and a set of mandatory and optional clusters to be implemented on a single
322 Zigbee endpoint. The term may also be used for an implementation or instance on an endpoint. In
323 this document, the unique identifier of a Zigbee Device maps to an OCF Device Type.

324 **3.1.8 Zigbee 3.0 Device**

325 Zigbee Device which is built on Zigbee 3.0 stack

326 **4 Document conventions and organization**

327 **4.1 Conventions**

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

332 **4.2 Notation**

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

335 Required (or shall or mandatory).

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

339 Recommended (or should).

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

347 Allowed (or allowed).

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

351 Conditionally allowed (CA)

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

354 Conditionally required (CR)

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

358 DEPRECATED

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

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

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

367 **5 Theory of Operation**

368 **5.1 Interworking Approach**

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

371 **5.2 Mapping Syntax**

372 **5.2.1 Introduction**

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

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

380 **5.2.2 General**

381 All statements are terminated with a carriage return.

382 **5.2.3 Value Assignment**

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

385 **5.2.4 Property Naming**

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

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

391 **5.2.5 Range**

392 The range on the OCF side is fixed.

393 **5.2.6 Arrays**

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

396 **5.2.7 Default Mapping**

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

400 **5.2.8 Conditional Mapping**

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

402 If "condition", then "mapping".

403 is applied.

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

405 **5.2.9 Method Invocation**

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

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

412 **6 Zigbee Translation**

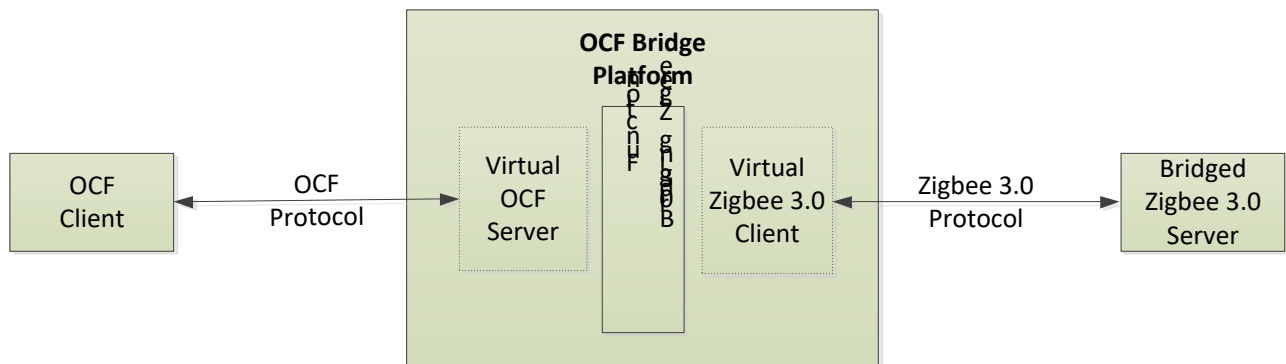
413 **6.1 Operational Scenarios**

414 The overall goal is to make Bridged Zigbee 3.0 Servers appear to OCF Clients as if they were
415 native OCF Servers in the local network or cloud environment

416 The mapping between the OCF data models and Zigbee Clusters is specified in 9. Programmatic
417 (i.e. On-the-fly) data model translation is not supported.

418 Figure 1 shows an overview of a Zigbee 3.0 Bridge Platform and its general topology. It exposes
419 Zigbee 3.0 Servers to OCF Clients. Each Bridged Zigbee 3.0 Server is represented as a Virtual
420 OCF Server. The Zigbee 3.0 Bridging Function supports Asymmetric bridging. The scope of this
421 document is the asymmetric bridging to expose the Zigbee Server to OCF. The asymmetric
422 bridging to expose an OCF Server to a Zigbee Client is out of scope.

423



424

425 **Figure 1 – OCF Zigbee Bridge Platform and Components**

426 **6.2 Requirements specific to Zigbee Bridging Function**

427 **6.2.1 Requirements specific to Zigbee**

428 This document refers to Zigbee 3.0 or higher. Zigbee 3.0 is built on Zigbee Pro 2015 or newer,
429 which enhances the IEEE 802.15.4 standard by adding a mesh network and security layers along
430 with an application framework. Low power support is not the scope of this document.

431 An OCF Zigbee Bridging Function shall act as a Zigbee Coordinator in network layer. A Zigbee
432 Coordinator is responsible for initiating and maintaining the devices on the network. An OCF
433 Zigbee Bridge Platform will act as Zigbee Client towards the Zigbee 3.0 Devices in the application
434 layer. Users can expect that a certified OCF Bridge Platform will be able to talk to Zigbee 3.0
435 Devices, without the user having to buy some other device.

436 **6.2.2 Exposing Zigbee 3.0 Servers to OCF Clients**

437 The nature of how Zigbee Devices are structured may be different than how an OCF Device is
438 structured. The mapping of the structure of a Zigbee device on an OCF Device is given by Table
439 1.

440 A Zigbee Server cluster may map to one or more OCF Resources. If a specific Zigbee Server
441 cluster has specific commands, one or more OCF Resources corresponding to the specific
442 command attributes may be additionally needed.

443 A Zigbee Attribute of a Zigbee Server cluster typically maps to an OCF Resource Property.
 444 However, in some special cases, multiple attributes are mapped to a single OCF Resource
 445 Property e.g., "CurrentX" and "CurrentY" of the Zigbee color control cluster map to the "csc"
 446 Property in the "oic.r.colour.csc" (Colour Space Coordinates) Resource because of the difference
 447 in the data types, i.e., "csc" is an array, but "CurrentX" and "CurrentY" map to a number.

448 Table 2 is a mapping example of this rule

449 **Table 1 – Translation Rule between Zigbee and OCF Data Models**

From Zigbee	mapping count	To OCF	mapping count
Zigbee Device	1	OCF Device	1
Zigbee Cluster	1	OCF Resource	n
Zigbee Attribute	1	OCF Resource Property	1

450 **Table 2 – Zigbee to OCF Mapping Example (Color Temperature Light)**

From Zigbee		To OCF	
Zigbee 3.0 Device	0x010c (Color Temperature Light)	OCF Device	oic.d.light (Light)
Zigbee Server Cluster	0x0006 (On/Off)	OCF Resource(s)	oic.r.switch.binary (Binary Switch)
	0x0300 (Color Control Cluster)		oic.r.colour.hs (Colour Hue and Saturation)
			oic.r.colour.csc (Colour Space Coordinates)
oic.r.colour.colourtemperature (Colour Temperature)			
Zigbee Attribute	0x0000 (OnOff of On/Off Cluster)	OCF Resource Property	value (of Binary Switch Resource)
	0x0003 (CurrentX of Color Control Cluster)		csc (of Colour Space Coordinates)
	0x0004 (CurrentY of Color Control Cluster)		

451 If a Zigbee 3.0 Device, Zigbee Server Cluster, Zigbee Attribute are enlisted in the well-defined set,
 452 the Bridging Function shall follow the requirements for translating it to an OCF Device, OCF
 453 Resource, or OCF Resource Property (i.e., "deep translation").

454 A Zigbee 3.0 Server Device maps to a single OCF Device Type. The OCF Device Type is provided
 455 by using the Device ID of the Zigbee 3.0 Server Device (The Device ID is allocated by the Zigbee
 456 Alliance and has the same meaning of the OCF Device Type). The Zigbee 3.0 Bridging Function
 457 has a table which includes the mapping information between the Zigbee Device ID and the OCF

458 Device Type. Based on the table, the Zigbee 3.0 Bridging Function finds the OCF Device Type
 459 according to the Zigbee Device ID.

460 A Zigbee Device includes one or more Zigbee Server Clusters. If a Zigbee Cluster maps to multiple
 461 OCF Resources, the Zigbee Cluster may be translated as a Resource with a Collection Resource
 462 Type. The resource mapping between Zigbee Server Cluster and OCF Resources is defined in 9
 463 for deep translation. The Zigbee 3.0 Bridging Function has a table which includes the mapping
 464 information between the identifier of Zigbee Cluster and OCF Resource Type(s). The Zigbee 3.0
 465 Bridging Function obtains the list of cluster identifiers after the Virtual Zigbee 3.0 Client and Zigbee
 466 3.0 Server Device are bound. Based on the table, the Zigbee 3.0 Bridging Function finds the OCF
 467 Resource Type(s) according to the identifier of Zigbee Cluster.

468 Since a Bridging Function knows all relationships between OCF Resources and Zigbee Server
 469 Clusters, the path component of URI can be free to choose. Maintaining relationship information
 470 and URI definition is implementation specific.

471 If a Zigbee operation fails, the Bridging Function send an appropriate OCF error response to the
 472 OCF Client. it construct an appropriate OCF error message (e.g., diagnostic payload if using CoAP)
 473 from the Zigbee enumerated status value and Zigbee enumerated status (if any), using the form
 474 "<error name>: <error message>", with the <error name> taken from the Zigbee Status Code field
 475 and the <error message> taken from the Zigbee enumerated status, and the error code for the
 476 OCF network set to an appropriate value.

477 **6.2.3 Translation for well-defined set**

478 If a Zigbee 3.0 Device, Zigbee Server Cluster, Zigbee Attribute are enlisted in the well-defined set,
 479 the Bridging Function shall follow the requirements for translating it to an OCF Device, OCF
 480 Resource, or OCF Resource Property (i.e., "deep translation"). Table 3 is the list of Zigbee 3.0
 481 devices and mandatory Zigbee Server Clusters with corresponding OCF devices and mandatory
 482 OCF Resources. Optional OCF Resources mapped with the specific Zigbee Server Clusters are
 483 enlisted in the well-defined set.

484 **Table 3 – Zigbee 3.0 Device & Cluster – OCF Device & Resource mapping**

Zigbee 3.0 Device Name (Device ID)	Zigbee 3.0 Mandatory Cluster	OCF Mandatory Resource Type	OCF Device Type ("rt")	OCF Device Name
On/off light (0x0100)	On/off	oic.r.switch.binary,	oic.d.light	Light
Color Temperature Light (0x010c)	On/off, Level Control, Color Control	oic.r.switch.binary,	oic.d.light	Light
Extended Color Light (0x010d)	On/off, Level Control, Color Control	oic.r.switch.binary,	oic.d.light	Light
Dimmable Light (0x0101)	On/off, Level Control	oic.r.switch.binary,	oic.d.light	Light

Zigbee 3.0 Device Name (Device ID)	Zigbee 3.0 Mandatory Cluster	OCF Mandatory Resource Type	OCF Device Type (“rt”)	OCF Device Name
Color Dimmable Light (0x0102)	On/off Level Control, Color Control	oic.r.switch.binary,	oic.d.light	Light
Temperature Sensor (0x0302)	Temperature Measurement	oic.r.temperature	oic.d.sensor	Generic Sensor
Thermostat (0x0301)	Thermostat	oic.r.temperature(2)	oic.d.thermostat	Thermostat
Window Covering Device (0x0202)	Window Covering	oic.r.openlevel	oic.d.blind	Blind
Smart Plug (0x0051)	On/off, Metering	oic.r.switch.binary,	oic.d.smartplug	Smart Plug
Mains Power Outlet (0x0009)	On/off	oic.r.switch.binary,	oic.d.smartplug	Smart Plug
On/off output (0x0002)	On/off	oic.r.switch.binary,	oic.d.smartplug	Smart Plug
IAS Zone (0x0402)	IAS Zone	oic.r.ias.zone	oic.d.sensor	Generic Sensor
Occupancy Sensor (0x0107)	Occupancy Sensing	oic.r.sensor.presence	oic.d.sensor	Generic Sensor

485 **6.2.4 Exposing a Zigbee 3.0 Server as a Virtual OCF Server**

486 Table 4 shows how OCF Platform properties, as specified in ISO/IEC 30118-1:2018, shall be
487 derived, typically from fields of Descriptor specified in Zigbee.

Table 4 – "oic.wk.p" Resource Type mapping

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory	From Zigbee 3.0 Field name	Zigbee 3.0 Description	Zigbee 3.0 Mandatory
Platform ID	pi	Unique identifier for the physical platform (UIUID); this shall be a UIUID in accordance with IETF RFC 4122. It is recommended that the UIUID be created using the random generation scheme (version 4 UIUID) specific in the RFC.	Y	(none)	Bridging Function should return a randomly-generated UIUID (Please see section 4.4 of IETF RFC 4122 for randomly-generated UIUID)	
Manufacturer Name	mnmn	Name of manufacturer (not to exceed 16 characters)	Y	Manufacturer name (in DefaultLanguage, truncated to 16 characters)	Name of the manufacturer as a ZigBee character string Defined in Basic Cluster	Y
Manufacturer Details Link (URL)	mnml	URL to manufacturer (not to exceed 32 characters)	N	(none)	(none)	N
Model Number	mnmo	Model number as designated by manufacturer	N	Model Identifier	Model number (or other identifier) assigned by the manufacturer as a ZigBee character string Defined in Basic Cluster	Y
Date of Manufacture	mndt	Manufacturing date of device	N	DateCode	Date of manufacturer of the device in international date notation according to ISO 8601, i.e., YYYYMMDD, Defined in Basic Cluster	N
Platform Version	mnpv	Version of platform – string (defined by manufacturer)	N	(none)	(none)	N
OS Version	mnos	Version of platform resident OS – string (defined by manufacturer)	N	(none)	(none)	N
Hardware Version	mnhw	Version of platform hardware	N	HWVersion	Version number of the hardware of the device. Defined in Basic Cluster	N
Firmware version	mnfv	Version of device firmware	N	(none)	(none)	N

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory	From Zigbee 3.0 Field name	Zigbee 3.0 Description	Zigbee 3.0 Mandatory
Support link	mnsI	URI that points to support information from manufacturer	N	ProductURL	Link to a web page containing specific product information Defined in Basic Cluster	N
SystemTime	st	Reference time for the device	N	(none)	(none)	N
Vendor ID	vid	Vendor defined string for the platform. The string is freeform and up to the vendor on what text to populate it.	N	(none)	(none)	N

489 Table 5 shows how OCF Device Properties, as specified in Table 20 in ISO/IEC 30118-1:2018,
490 shall be derived, typically from fields of Descriptor or Attributes of Basic cluster specified in Zigbee
491 and Zigbee Cluster Library Specification, respectively.

492 As specified in ISO/IEC 30118-2:2019, the value of the “di” Property of OCF Devices (including
493 Virtual OCF Devices) shall be established as part of Onboarding of that Virtual OCF Device.

494 **Table 5 – "oic.wk.d" Resource Type mapping**

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory	From Zigbee 3.0 Field name	Zigbee 3.0 Description	Zigbee 3.0 Mandatory
(Device) Name	n	Human friendly name For example, “Bob’s Thermostat”	Y	User description if it exists, else Model Name if it exists, else translate Application Device Identifier (=Device ID) to Human friendly name by using Application Device Identifier value/description table	User description : Information that allows the user to identify the device using a user-friendly character string, such as “Bedroom TV” Defined in User Descriptor Model Name : character string representing the name of the manufacturer’s model of the device Defined in Complex Descriptor Application Device Identifier: device description supported on this endpoint Cluster Defined in Simple Descriptor	User description: N Model Name: N Application Device Identifier: Y
Spec Version	icv	Spec version of the core specification this device is implemented to,	Y	(none)	Spec version of the core specification that the Bridging Platform	

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory	From Zigbee 3.0 Field name	Zigbee 3.0 Description	Zigbee 3.0 Mandatory
		The syntax is "core.major.minor"]			implements should return its own value	
Device ID	di	Unique identifier for Device. This value shall be as defined in OCF Security Specification for DeviceID.	Y	(none)	Use as defined in the ISO/IEC 30118-2:2019	
Protocol-Independent ID	piid	Unique identifier for OCF Device (UUID)	Y	(none)	Bridging Function should return a randomly-generated UUID (Please see section 4.4 of IETF RFC 4122 for randomly-generated UUID)	
Data Model Version	dmv	Spec version(s) of the vertical specifications this device data model is implemented to. The syntax is a comma separated list of "<vertical>.major.minor"]. <vertical> is the name of the vertical (i.e. sh for Smart Home)	Y	(none)	Bridging Function should return its own value.	
Localized Descriptions	ld	Detailed description of the Device, in one or more languages. This property is an array of objects where each object has a "language" field (containing an RFC 5646 language tag) and a "value" field containing the device description in the indicated language.	N	(none)	Zigbee provides Language and Character Set field only which specifies the language and character set used by the character strings by using ISO 639-1 language code	

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory	From Zigbee 3.0 Field name	Zigbee 3.0 Description	Zigbee 3.0 Mandatory
Software Version	sv	Version of the device software.	N	ApplicationVersion	Version number of the application software contained in the device. Defined in Basic Cluster	Y
Manufacturer Name	dmn	Name of manufacturer of the Device, in one or more languages. This property is an array of objects where each object has a "language" field (containing an RFC 5646 language tag) and a "value" field containing the manufacturer name in the indicated language.	N	Manufacturer name	Name of the manufacturer as a ZigBee character string Defined in Basic Cluster	Y
Model Number	dmno	Model number as designated by manufacturer.	N	Model Identifier	Model number (or other identifier) assigned by the manufacturer as a ZigBee character string Defined in Basic Cluster	Y

495 Table 6 shows how OCF Device Configuration properties, as specified in Table 15 in ISO/IEC
496 30118-1:2018 shall be derived.

Table 6 – "oic.wk.con" Resource Type mapping

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory	From 3.0 Zigbee Field name	Zigbee 3.0 Description	Zigbee 3.0 Mandatory
(Device) Name	n	Human friendly name For example, "Bob's Thermostat"	Y	User description if it exists, else Model Name if it exists, else translate Application Device Identifier (=Device ID)to Human friendly name by using Application Device Identifier value/description table	User description : Information that allows the user to identify the device using a user-friendly character string, such as "Bedroom TV" Defined in User Descriptor Model Name : character string representing the name of the manufacturer's model of the device Defined in Complex Descriptor Application Device Identifier: device description supported on this endpoint Cluster Defined in Simple Descriptor	User description: N Model Name: N Application Device Identifier: Y
Location	loc	Provides location information where available.	N	(none)	(none)	
Location Name	locn	Human friendly name for location For example, "Living Room".	N	(none)	(none)	
Currency	c	Indicates the currency that is used for any monetary transactions	N	(none)	(none)	
Region	r	Free form text Indicating the current region in which the device is located geographically. The free form text shall not start with a quote (").	N	(none)	(none)	
Localized Names	ln	Human-friendly name of the Device, in one or more languages. This property is an array of objects where each object has a "language" field (containing an RFC 5646 language tag) and a "value" field containing the	N	User description if it exists, else Model Name if it exists, else translate Application Device Identifier (=Device ID)to Human friendly name	User description : Information that allows the user to identify the device using a user-friendly character string, such as "Bedroom TV" Defined in User Descriptor Model Name : character string representing the	User description: N Model Name: N Application Device Identifier: Y

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory	From Zigbee 3.0 Field name	Zigbee 3.0 Description	Zigbee 3.0 Mandatory
		device name in the indicated language. If this property and the Device Name (n) property are both supported, the Device Name (n) value shall be included in this array.		by using Application Device Identifier value/description table	name of the manufacturer's model of the device Defined in Complex Descriptor Application Device Identifier: device description supported on this endpoint Cluster Defined in Simple Descriptor	
Default Language	dl	The default language supported by the Device, specified as an RFC 5646 language tag. By default, clients can treat any string property as being in this language unless the property specifies otherwise.	N	ISO 639-1 language code (if it exists, else property is absent)	Language used for character strings.	N

498

499 7 Device Type Mapping

500 7.1 Introduction

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

502 7.2 Zigbee Device Types to OCF Device Types

503 Table 7 captures the equivalency mapping between Zigbee defined Device Types (Please see
504 reference Zigbee Cluster Library Specification) and OCF defined Device Types (please see
505 reference ISO/IEC 30118-5:2019).

506

Table 7 – 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

507 **8 Resource to ZigBee Cluster Equivalence**

508 **8.1 Introduction**

509 This clause introduces new Resource Types for mapping between Zigbee Clusters and OCF
510 Resources and lists the complete set of applicable Zigbee Clusters and equivalent OCF Resource
511 Type(s) in clause 8.2

512 **8.2 Zigbee Clusters to OCF Resources**

513 **8.2.1 Introduction**

514 Table 8 captures the equivalency mapping between Zigbee defined Clusters (see Zigbee Cluster
515 Library Specification) and OCF defined Resource Types (see ISO/IEC 30118-4:2019). Detailed
516 Property by Property mappings are provided in clause 8.1.

517 clause 9 captures the mappings for mandatory server clusters for Zigbee 3.0 devices

Table 8 – 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

519

8.2.2 On/off

521 The APIs with "zcl.onoff" define the mapping between an instance of an OCF Binary Switch
522 Resource and the Zigbee On/off Cluster. In clause 9.15 a RETRIEVE on an OCF Binary Switch
523 Resource maps to a general Read command on a Zigbee On/off Cluster. The value of Zigbee
524 Attribute in Zigbee On/off Cluster is retrieved via the general Read command and mapped with the
525 value of OCF Property in OCF Binary Switch Resource. In clause 9.14 an UPDATE on a Binary
526 Switch maps to a command invocation on either "on()" command or "off()" command of Zigbee
527 On/off Cluster. "value = true" maps to "on()", "value = false" maps to "off()" of Zigbee On/off Cluster.

8.2.3 Level Control

528 The APIs with "zcl.levelcontrol" define the mapping between an instance of an OCF Dimming
529 Resource and the Zigbee Level Control Cluster. In clause 9.12, a RETRIEVE on an OCF Dimming
530 Resource maps to a general Read command on a Zigbee Level Control Cluster. The value of
531

532 Zigbee Attribute in Zigbee Level Control Cluster is retrieved via the general Read command and
533 mapped with the value of OCF Property in OCF Dimming Resource. In clause 9.11, an UPDATE
534 on a "dimmingSetting" maps to a command invocation on "movetolevel(level,transitiontime=0)" of
535 Zigbee Level Control Cluster.

536 **8.2.4 Color Control**

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

540 The APIs with "zcl.colorcontrol_hs" define the mapping between an instance of OCF Hue and
541 Saturation Resources and the Zigbee Color Control Cluster. In clause 9.8, a RETRIEVE on an are
542 OCF Hue and Saturation Resource maps to a general Read command on a Zigbee Color Control
543 Cluster. The values of Zigbee Attributes in Zigbee Color Control Cluster are retrieved via the
544 general Read command and mapped with those of OCF Properties in OCF Hue and Saturation
545 Resource. In clause 9.7, an UPDATE on OCF Colour Hue and Saturation Resource maps to a
546 command invocation on "movetohueandsaturation(hue,saturation,transitiontime=0)" of Zigbee
547 Color Control Cluster.

548 The APIs with "zcl.colorcontrol_csc" define the mapping between an instance of OCF Colour Space
549 Coordinate Resource and the Zigbee Color Control Cluster. In clause 9.4, a RETRIEVE on an OCF
550 Colour Space Coordinate Resource maps to a general Read command on a Zigbee Color Control
551 Cluster. The values of Zigbee Attributes in Zigbee Color Control Cluster are retrieved via the
552 general Read command and mapped with those of OCF Properties in OCF Colour Space
553 Coordinate Resource. In clause 9.3, an UPDATE on OCF Colour Space Coordinate Resource
554 maps to a command invocation on "movetocolor(colorx,colory,transitiontime=0)" of Zigbee Color
555 Control Cluster.

556 The APIs with "zcl.colorcontrol_ct" define the mapping between an instance of OCF Colour
557 Temperature Resource and the Zigbee Color Control Cluster. In clause 9.5, a RETRIEVE on an
558 OCF Colour Temperature Resource maps to a general Read command on a Zigbee Color Control
559 Cluster. The values of Zigbee Attributes in Zigbee Color Control Cluster are retrieved via the
560 general Read command and mapped with those of OCF Properties in OCF Colour Temperature
561 Resource. In clause 9.6, an UPDATE on OCF Colour Temperature Resource maps to a command
562 invocation on "movetocolortemperature(colortemperature,transitiontime=0)" of Zigbee Color
563 Control Cluster.

564 **8.2.5 Thermostat**

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

568 The API with "zcl.thermostat_currenttemperature" defines the mapping between an instance of
569 OCF Temperature Resource and the Zigbee Thermostat Cluster for sensor. In clause 9.18, a
570 RETRIEVE on an OCF Temperature Resource maps to a general Read command on a Zigbee
571 Thermostat Cluster. The value of Zigbee Attribute in Zigbee Thermostat Cluster is retrieved via the
572 general Read command and mapped with the value of OCF Property in OCF Temperature
573 Resource. The value represents the current temperature.

574 The API with "zcl.thermostat_heat" defines the mapping between an instance of OCF Temperature
575 Resource and the Zigbee Thermostat Cluster for heater. In clause 9.19, an UPDATE on
576 "temperature" of OCF Temperature Resource maps to "setpointraiselower(mode=heat mode,
577 amount)" on a Zigbee Thermostat Cluster.

578 The API with "zcl.thermostat_cool" defines the mapping between an instance of OCF Temperature
579 Resource and the Zigbee Thermostat Cluster for cooler. In clause 9.17, an UPDATE on

580 "temperature" of OCF Temperature Resource maps to "setpointraiselower(mode=cool mode,
581 amount)" on a Zigbee Thermostat Cluster.

582 **8.2.6 Window Covering**

583 The APIs with "zcl.windowcovering" define the mapping between 5 instances of OCF Resources
584 and the Zigbee Window Covering Cluster. The 5 instances of OCF Resources are the instance of
585 OCF Window Covering Resource and the 4 instances of OCF Open Level Resources. The 4
586 instances of OCF Open Level Resources are for lift level with percentage scale, lift level with
587 centimetre scale, tilt level with percentage scale, tilt level with centimetre scale.

588 The API with "zcl.windowcovering_conf" defines the mapping between an instance of OCF Window
589 Covering Resource and the Zigbee window Covering Cluster. In clause 9.21, a RETRIEVE on an
590 OCF Window Covering Resource maps to a general Read command on a Zigbee Window Covering
591 Cluster. The values of Zigbee Attributes in Zigbee Window Covering Cluster is retrieved via the
592 general Read command and mapped with the value of OCF Property in OCF Window Covering
593 Resource. In clause 9.20, an UPDATE on OCF Window Covering Resource maps to a general
594 Write command on a Zigbee Window Covering Cluster.

595 The API with "zcl.windowcovering_liftpercentage" defines the mapping between an instance of
596 OCF Open Level Resource and the Zigbee window Covering Cluster for lift with percentage scale.
597 In clause 9.23, a RETRIEVE on an OCF Open Level Resource maps to a general Read command
598 on "CurrentPositionLiftPercentage" of Zigbee Window Covering Cluster. The value of Zigbee
599 Attribute in Zigbee Window Covering Cluster is retrieved via the general Read command and
600 mapped with the value of OCF Property in OCF Open Level Resource. In clause 9.22, an UPDATE
601 on OCF Open Level Resource maps to "gotoliftpercentage(percentageliftvalue)" on a Zigbee
602 Window Covering Cluster.

603 The API with "zcl.windowcovering_liftposition" defines the mapping between an instance of OCF
604 Open Level Resource and the Zigbee window Covering Cluster for lift with centimetre scale. In
605 clause 9.25, a RETRIEVE on an OCF Open Level Resource maps to a general Read command on
606 "CurrentPosition-Lift" of Zigbee Window Covering Cluster. The value of Zigbee Attribute in Zigbee
607 Window Covering Cluster is retrieved via the general Read command and mapped with the value
608 of OCF Property in OCF Open Level Resource. In clause 9.24, an UPDATE on OCF Open Level
609 Resource maps to "gotoliftvalue(liftvalue)" on a Zigbee Window Covering Cluster.

610 The API with "zcl.windowcovering_tiltpercentage" defines the mapping between an instance of
611 OCF Open Level Resource and the Zigbee window Covering Cluster for tilt with percentage scale.
612 In clause 9.27, a RETRIEVE on an OCF Open Level Resource maps to a general Read command
613 on "CurrentPositionTiltPercentage" of Zigbee Window Covering Cluster. The value of Zigbee
614 Attribute in Zigbee Window Covering Cluster is retrieved via the general Read command and
615 mapped with the value of OCF Property in OCF Open Level Resource. In clause 9.26, an UPDATE
616 on OCF Open Level Resource maps to "gototiltpercentage(percentagetiltvalue)" on a Zigbee
617 Window Covering Cluster.

618 The API with "zcl.windowcovering_tiltposition" defines the mapping between an instance of OCF
619 Open Level Resource and the Zigbee window Covering Cluster for tilt with centimetre scale. In
620 clause 9.29, a RETRIEVE on an OCF Open Level Resource maps to a general Read command on
621 "CurrentPosition-Tilt" of Zigbee Window Covering Cluster. The value of Zigbee Attribute in Zigbee
622 Window Covering Cluster is retrieved via the general Read command and mapped with the value
623 of OCF Property in OCF Open Level Resource. In clause 9.28, an UPDATE on OCF Open Level
624 Resource maps to "gototiltvalue(tiltvalue)" on a Zigbee Window Covering Cluster.

625 **8.2.7 Temperature Measurement**

626 The API with "zcl.temperaturemeasurement" defines the mapping between an instance of an OCF
627 Temperature Resource and the Zigbee Temperature Measurement Cluster for sensor. In clause
628 9.16, a RETRIEVE on an OCF Temperature Resource maps to a general Read command on a

629 Zigbee Temperature Measurement Cluster. The value of Zigbee Attribute in Zigbee Temperature
 630 Measurement Cluster is retrieved via the general Read command and mapped with the value of
 631 OCF Property in OCF Temperature Resource. The value represents the current temperature.

632 8.2.8 Occupancy Sensing

633 The API with "zcl.occupancysensing" defines the mapping between an instance of an OCF
 634 Presence Sensor Resource and the Zigbee Occupancy Sensing Cluster. In clause 9.13, a
 635 RETRIEVE on an OCF Presence Sensor Resource maps to a general Read command on a Zigbee
 636 Occupancy Sensing Cluster. The value of Zigbee Attribute in Zigbee Occupancy Sensing Cluster
 637 is retrieved via the general Read command and mapped with the value of OCF Property in OCF
 638 Presence Sensor.

639 8.2.9 IAS Zone

640 The API with "zcl.iaszone" defines the mapping between an instance of an OCF IAS Zone
 641 Resource and the Zigbee IAS Zone Cluster. In clause 9.10, a RETRIEVE on an IAS Zone Resource
 642 maps to a general Read command on a Zigbee IAS Zone Cluster. The values of Zigbee Attributes
 643 in Zigbee IAS Zone Cluster are retrieved via the general Read command and mapped with those
 644 of OCF Properties in OCF IAS Zone Resource. In clause 9.9, an UPDATE on OCF IAS Zone
 645 Resource maps to a general Write command on a Zigbee IAS Zone Cluster.

646 9 Detailed Mapping APIs

647 9.1 below

648 9.2 Introduction

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

652 The derived model definitions presented in clause 9 are formatted for readability, and so may
 653 appear to have extra line breaks.

654 9.3 Color Control Cluster - Color Space - Control

655 9.3.1 Derived model

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

657 9.3.2 Property definition

658 Table 9 provides the detailed per Property mapping for "zcl.colorcontrol_csc.control.movetocolor".

659 **Table 9 – The Property mapping for "zcl.colorcontrol_csc.control.movetocolor".**

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

660 Table 10 provides the details of the Properties that are part of
 661 "zcl.colorcontrol_csc.control.movetocolor".

Table 10 – The Properties of "zcl.colorcontrol_csc.control.movetocolor".

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

663 9.3.3 Derived model definition

```

664 {
665   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_csc.control.json#",
666   "$schema": "http://json-schema.org/draft-04/schema#",
667   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
668   "title": "Color Control Cluster - Color Space - Control",
669   "definitions": {
670     "zcl.colorcontrol_csc.control.movetocolor": {
671       "properties": {
672         "colorx": {
673           "type": "number",
674           "description": "Move to certain value(s) of color coordinates as fast as possible with
675 transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
676           "x-ocf-conversion": {
677             "x-ocf-alias": "oic.r.colour.csc",
678             "x-from-ocf": [
679               "colorx =ocf.csc[0]*65536 & transitiontime=0",
680               "zcl.command.colorcontrol::movetocolor(colorx,colory,transitiontime).",
681             ],
682             "x-to-ocf": [
683               "N/A"
684             ]
685           }
686         },
687         "colory": {
688           "type": "number",
689           "description": "Move to certain value(s) of color coordinates as fast as possible with
690 transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
691           "x-ocf-conversion": {
692             "x-ocf-alias": "oic.r.colour.csc",
693             "x-from-ocf": [
694               "colory= ocf.csc[1]*65536 & transitiontime=0",
695               "zcl.command.colorcontrol::movetocolor(colorx,colory,transitiontime).",
696             ],
697             "x-to-ocf": [
698               "N/A"
699             ]
700           }
701         }
702       }
703     }
704   },
705   "type": "object",
706   "allOf": [
707     {"$ref": "#/definitions/zcl.colorcontrol_csc.control.movetocolor"}
708   ]

```

709 }
710

711 9.4 Color Control Cluster - Color Space - Information

712 9.4.1 Derived model

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

714 9.4.2 Property definition

715 Table 11 provides the detailed per Property mapping for "zcl.colorcontrol_csc.info".

716 **Table 11 – 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

717 Table 12 provides the details of the Properties that are part of "zcl.colorcontrol_csc.info".

718 **Table 12 – 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

719 9.4.3 Derived model definition

```
720 {  
721   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_csc.info.json#",  
722   "$schema": "http://json-schema.org/draft-04/schema#",  
723   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",  
724   "title": "Color Control Cluster - Color Space - Information",  
725   "definitions": {  
726     "zcl.colorcontrol_csc.info": {  
727       "type": "object",  
728       "properties": {  
729         "currentX": {  
730           "type": "integer",  
731           "description": "current value of the normalized chromaticity value x, as defined in the  
732 CIE xy Color Space",  
733           "x-ocf-conversion": {  
734             "x-ocf-alias": "oic.r.colour.csc",  
735             "x-to-ocf": [  
736               "ocf.csc[0] = currentX/65536"  
737             ],  
738             "x-from-ocf": [  
739               "N/A"  
740             ]  
741           }  
742         },  
743         "currentY": {  
744           "type": "integer",
```

```

745         "description": "current value of the normalized chromaticity value y, as defined in the
746 CIE xy Color Space",
747         "x-ocf-conversion": {
748             "x-ocf-alias": "oic.r.colour.csc",
749             "x-to-ocf": [
750                 "ocf.csc[1] = currentY/65536"
751             ],
752             "x-from-ocf": [
753                 "N/A"
754             ]
755         }
756     }
757 }
758 },
759 },
760 "type": "object",
761 "allOf": [
762     {"$ref": "#/definitions/zcl.colorcontrol_csc.info"}
763 ],
764 "required": ["currentx", "currenty"]
765 }
766

```

767 **9.5 Color Control Cluster - Color Temperature - Information**

768 **9.5.1 Derived model**

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

770 **9.5.2 Property definition**

771 Table 13 provides the detailed per Property mapping for
772 "zcl.colorcontrol_ct.control.movetocolortemperature".

773 **Table 13 – The Property mapping for**
774 **"zcl.colorcontrol_ct.control.movetocolortemperature".**

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

775 Table 14 provides the details of the Properties that are part of
776 "zcl.colorcontrol_ct.control.movetocolortemperature".

777 **Table 14 – The Properties of "zcl.colorcontrol_ct.control.movetocolortemperature".**

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

778 **9.5.3 Derived model definition**

```

779 {
780     "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_ct.control.json#",
781     "$schema": "http://json-schema.org/draft-04/schema#",
782     "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",

```

```

783     "title": "Color Control Cluster - Color Temperature - Information",
784     "definitions": {
785       "zcl.colorcontrol_ct.control.movetocolortemperature": {
786         "properties": {
787           "colortemperature": {
788             "type": "integer",
789             "description": "Move to certain value of colortemperature as fast as possible with
790 transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
791             "x-ocf-conversion": {
792               "x-ocf-alias": "oic.r.colour.colourtemperature",
793               "x-from-ocf": [
794                 "colourtemperature=ocf.ct & transitiontime=0",
795                 "zcl.command.colorcontrol::movetocolortemperature(colortemperature,transitiontime)"
796               ],
797               "x-to-ocf": [
798                 "N/A"
799               ]
800             }
801           }
802         }
803       },
804     },
805     "type": "object",
806     "allOf": [
807       {"$ref": "#/definitions/zcl.colorcontrol_ct.control.movetocolortemperature"}
808     ]
809   }
810 }

```

811 9.6 Color Control Cluster - Color Temperature - Information

812 9.6.1 Derived model

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

814 9.6.2 Property definition

815 Table 15 provides the detailed per Property mapping for "zcl.colorcontrol_ct.info".

816 **Table 15 – The Property mapping for "zcl.colorcontrol_ct.info".**

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

817 Table 16 provides the details of the Properties that are part of "zcl.colorcontrol_ct.info".

818 **Table 16 – The Properties of "zcl.colorcontrol_ct.info".**

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

819 9.6.3 Derived model definition

```
820 {
821   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_ct.info.json#",
822   "$schema": "http://json-schema.org/draft-04/schema#",
823   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
824   "title": "Color Control Cluster - Color Temperature - Information",
825   "definitions": {
826     "zcl.colorcontrol_ct.info": {
827       "type": "object",
828       "properties": {
829         "colortemperaturemired": {
830           "type": "integer",
831           "description": "Scaled inverse of the current value of the color temperature",
832           "x-ocf-conversion": {
833             "x-ocf-alias": "oic.r.colour.colourtemperature",
834             "x-to-ocf": [
835               "ocf.ct = colortemperaturemired"
836             ],
837             "x-from-ocf": [
838               "N/A"
839             ]
840           }
841         },
842         "colortemphysicalmin": {
843           "type": "integer",
844           "description": "minimum mired value supported by the hardware",
845           "x-ocf-conversion": {
846             "x-ocf-alias": "oic.r.colour.colourtemperature",
847             "x-to-ocf": [
848               "ocf.range[0] = colortemphysicalmin"
849             ],
850             "x-from-ocf": [
851               "N/A"
852             ]
853           }
854         },
855         "colorphysicalmax": {
856           "type": "integer",
857           "description": "maximum mired value supported by the hardware",
858           "x-ocf-conversion": {
859             "x-ocf-alias": "oic.r.colour.colourtemperature",
860             "x-to-ocf": [
861               "ocf.range[1] = colorphysicalmax"
862             ],
863             "x-from-ocf": [
864               "N/A"
865             ]
866           }
867         }
868       }
869     }
870   },
871   "type": "object",
872   "allOf": [
873     {"$ref": "#/definitions/zcl.colorcontrol_ct.info"}
874   ],
875   "required": ["colortemperaturemired", "colortemphysicalmin", "colortempphysicalmax" ]
876 }
877
```

878 9.7 Color Control Cluster - Hue and Saturation - Control

879 9.7.1 Derived model

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

881 9.7.2 Property definition

882 Table 17 provides the detailed per Property mapping for
883 "zcl.colorcontrol_hs.control.movetohueandsaturation".

884
885

Table 17 – The Property mapping for "zcl.colorcontrol_hs.control.movetohueandsaturation".

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

886 Table 18 provides the details of the Properties that are part of
887 "zcl.colorcontrol_hs.control.movetohueandsaturation".

888 **Table 18 – The Properties of "zcl.colorcontrol_hs.control.movetohueandsaturation".**

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

889 **9.7.3 Derived model definition**

```

890 {
891   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_hs.control.json#",
892   "$schema": "http://json-schema.org/draft-04/schema#",
893   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
894   "title": "Color Control Cluster - Hue and Saturation - Control",
895   "definitions": {
896     "zcl.colorcontrol_hs.control.movetohueandsaturation": {
897       "properties": {
898         "hue": {
899           "type": "integer",
900           "description": "Move to certain value(s) of hue or saturation or both as fast as possible
901 with transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
902           "x-ocf-conversion": {
903             "x-ocf-alias": "oic.r.colour.hs",
904             "x-from-ocf": [
905               "hue=ocf.hue/360 * 254 & transitiontime=0",
906               "zcl.command.colorcontrol::movetohueandsaturation(hue,saturation,transitiontime)"
907             ],
908             "x-to-ocf": [
909               "N/A"
910             ]
911           }
912         },
913         "saturation": {

```

```

914         "type": "integer",
915         "description": "Move to certain value(s) of hue or saturation or both as fast as possible
916 with transitiontime=0. transitiontime is set by Zigbee 3.0 translator.",
917         "x-ocf-conversion": {
918             "x-ocf-alias": "oic.r.colour.hs",
919             "x-from-ocf": [
920                 "saturation=ocf.saturation & transitiontime=0",
921                 "zcl.command.colorcontrol::movetohueandsaturation(hue,saturation,transitiontime)"
922             ],
923             "x-to-ocf": [
924                 "N/A"
925             ]
926         }
927     }
928 }
929 }
930 },
931 "type": "object",
932 "allOf": [
933     {"$ref": "#/definitions/zcl.colorcontrol_hs.control.movetohueandsaturation"}
934 ]
935 }
936

```

937 9.8 Color Control Cluster - Hue and Saturation - Information

938 9.8.1 Derived model

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

940 9.8.2 Property definition

941 Table 19 provides the detailed per Property mapping for "zcl.colorcontrol_hs.info".

942 **Table 19 – 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

943 Table 20 provides the details of the Properties that are part of "zcl.colorcontrol_hs.info".

944 **Table 20 – 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

945 9.8.3 Derived model definition

```

946 {
947     "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.colorcontrol_hs.info.json#",
948     "$schema": "http://json-schema.org/draft-04/schema#",
949     "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
950     "title": "Color Control Cluster - Hue and Saturation - Information",
951     "definitions": {
952         "zcl.colorcontrol_hs.info": {
953             "type": "object",
954             "properties": {
955                 "currenthue": {

```

```

956     "type": "integer",
957     "description": "current hue value of the light",
958     "x-ocf-conversion": {
959       "x-ocf-alias": "oic.r.colour.hs",
960       "x-to-ocf": [
961         "ocf.hue = currenthue/254 * 360"
962       ],
963       "x-from-ocf": [
964         "N/A"
965       ]
966     },
967   },
968   "currentsaturation": {
969     "type": "integer",
970     "description": "current saturation value of the light",
971     "x-ocf-conversion": {
972       "x-ocf-alias": "oic.r.colour.hs",
973       "x-to-ocf": [
974         "ocf.saturation = currentsaturation & maximumsaturation=254"
975       ],
976       "x-from-ocf": [
977         "N/A"
978       ]
979     }
980   }
981 }
982 }
983 },
984 "type": "object",
985 "allOf": [
986   {"$ref": "#/definitions/zcl.colorcontrol_hs.info"}
987 ],
988 "required": ["currenthue", "currentsaturation"]
989 }
990

```

9.9 IAS Zone Cluster - Control

9.9.1 Derived model

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

9.9.2 Property definition

Table 21 provides the detailed per Property mapping for "zcl.iaszone.control".

Table 21 – The Property mapping for "zcl.iaszone.control".

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

Table 22 provides the details of the Properties that are part of "zcl.iaszone.control".

Table 22 – The Properties of "zcl.iaszone.control".

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

9.9.3 Derived model definition

```

1000 {
1001   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.iaszone.control.json#",
1002   "$schema": "http://json-schema.org/draft-04/schema#",

```

```

1003 "description" : "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1004 "title": "IAS Zone Cluster - Control",
1005 "definitions": {
1006   "zcl.iaszone.control": {
1007     "properties": {
1008       "currentzonesensitivitylevel": {
1009         "type": "integer",
1010         "description": "Set a sensitivity level of IAS Zone",
1011         "x-ocf-conversion": {
1012           "x-ocf-alias": "oic.r.ias.zone",
1013           "x-from-ocf": [
1014             "currentzonesensitivitylevel = ocf.currentzonesensitivitylevel",
1015             "zcl.command.general::write(currentzonesensitivitylevel)"
1016           ],
1017           "x-to-ocf": [
1018             "N/A"
1019           ]
1020         }
1021       }
1022     }
1023   },
1024   "type": "object",
1025   "allOf": [
1026     {"$ref": "#/definitions/zcl.iaszone.control"}
1027   ]
1028 }
1029 }
1030

```

1031 **9.10 IAS Zone Cluster - Information**

1032 **9.10.1 Derived model**

1033 The derived model: "zcl.iaszone.info".

1034 **9.10.2 Property definition**

1035 Table 23 provides the detailed per Property mapping for "zcl.iaszone.info".

1036 **Table 23 – The Property mapping for "zcl.iaszone.info".**

Zigbee Property name	OCF Resource	To OCF	From OCF
zoneID	oic.r.iaszone	ocf.zoneid=zoneID	N/A
numberofzonesensitivitylevels_supported	oic.r.iaszone	ocf.numzonesensitivitylevels_supported=numberofzonesensitivitylevels_supported	N/A
zonestate	oic.r.iaszone	if zonestate=0x00, ocf.zonestate=false if zonestate=0x01, ocf.zonestate=true	N/A
IAS_CIE_address	oic.r.iaszone	ocf.iascieaddress= IAS_CIE_address	N/A
zonetype	oic.r.iaszone	if zonetype=0x0000, ocf.zonetype=Standard CIE if zonetype=0x000d, ocf.zonetype=Motion sensor if zonetype=0x0015, ocf.zonetype=Contact switch if zonetype=0x0028, ocf.zonetype=Fire sensor if zonetype=0x002a, ocf.zonetype=Water sensor if zonetype=0x002b, ocf.zonetype=Carbon Monoxide (CO) sensor if zonetype=0x002c, ocf.zonetype=Personal emergency device if zonetype=0x002d, ocf.zonetype=Vibration/Movement sensor if	N/A

		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	
zonestatus	oic.r.iasz one	if zonetype=0x0000 & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['']if zonetype=0x0000 & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['system']if zonetype=0x0000 & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']if zonetype=0x0000 & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['']if zonetype=0x000d & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['']if zonetype=0x000d & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['intrusion']if zonetype=0x000d & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']if zonetype=0x000d & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['presence']if zonetype=0x000d & zonestatus=xxxxxxxxxxxx11, 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','2n dportalopenclose']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 &	N/A

		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 & zonestatus=xxxxxxxxxxxx11, ocf.zonestatus.alarms=['fall','emergencybutton'] if zonetype=0x002d & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['']if zonetype=0x002d & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['movement']if zonetype=0x002d & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']if zonetype=0x002d & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['vibration']if zonetype=0x002d & zonestatus=xxxxxxxxxxxx11, ocf.zonestatus.alarms=['movement','vibration']if zonetype=0x010f & zonestatus=xxxxxxxxxxxx0, ocf.zonestatus.alarms=['']if zonetype=0x010f & zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=['panic']if zonetype=0x010f & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']if zonetype=0x010f & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=['emergency']if zonetype=0x010f &	
--	--	--	--

		<pre> zonestatus=xxxxxxxxxxx11, ocf.zonestatus.alarms=['panic','emergency']if zonetype=0x0115 & zonestatus=xxxxxxxxxxx0, ocf.zonestatus.alarms=['']if zonetype=0x0115 & zonestatus=xxxxxxxxxxx1, ocf.zonestatus.alarms=['panic']if zonetype=0x0115 & zonestatus=xxxxxxxxxxx0x, ocf.zonestatus.alarms=['']if zonetype=0x0115 & zonestatus=xxxxxxxxxxx1x, ocf.zonestatus.alarms=['emergency']if zonetype=0x0115 & zonestatus=xxxxxxxxxxx11, ocf.zonestatus.alarms=['panic','emergency']if zonetype=0x021d & zonestatus=xxxxxxxxxxx0, 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=xxxxxxxxxxx11, ocf.zonestatus.alarms=['panic','emergency']if zonetype=0x0225 & zonestatus=xxxxxxxxxxx0, 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 & </pre>	
--	--	--	--

		<pre> zonestatus=xxxxxxxxxxxx1, ocf.zonestatus.alarms=[']if zonetype=0xffff & zonestatus=xxxxxxxxxxxx0x, ocf.zonestatus.alarms=[']if zonetype=0xffff & zonestatus=xxxxxxxxxxxx1x, ocf.zonestatus.alarms=[']if zonestatus=xxxxxxxxxxxx0xx, ocf.zonestatus.tamper=falseif zonestatus=xxxxxxxxxxxx1xx, ocf.zonestatus.tamper=trueif zonestatus=xxxxxxxxxxxx0xxx, ocf.zonebattery.charge=100 & ocf.zonebattery.lowbattery=falseif zonestatus=xxxxxxxxxxxx1xxx, ocf.zonebattery.charge=100 & ocf.zonebattery.lowbattery=trueif zonestatus=xxxxxxxx00xxxx, 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=xxxxxxxx0xxxxxx, ocf.zonestatus.fault=falseif zonestatus=xxxxxxxx1xxxxxx, ocf.zonestatus.fault=trueif zonestatus=xxxxxx0xxxxxxx, ocf.zonepowersource.powerSources=['AC (Mains) Power'] & ocf.zonepowersource.sourcefault=falseif zonestatus=xxxxxx1xxxxxxx, ocf.zonepowersource.powerSources=['AC (Mains) Power'] & ocf.zonepowersource.sourcefault=trueif zonestatus=xxxxx0xxxxxxx, ocf.zonestatus.test=falseif zonestatus=xxxxx1xxxxxxx, ocf.zonestatus.test=trueif zonestatus=xxxx0xxxxxxx, ocf.zonepowersource.powerSources=['Internal Battery'] & oic.r.ias.zone.zonebattery.defect=false & oic.r.ias.zone.zonebattery.charge=100.if zonestatus=xxxx1xxxxxxx, oic.r.ias.zone.zonepowersource.powerSources =['Internal Battery'] & oic.r.ias.zone.zonebattery.defect=true & oic.r.ias.zone.zonebattery.charge=100. </pre>		
currentzonesensitivitylevel	oic.r.iaszone	ocf.currentzonesensitivitylevel	=	N/A

1037 Table 24 provides the details of the Properties that are part of "zcl.iaszone.info".

Table 24 – The Properties of "zcl.iaszone.info".

Zigbee Property name	Type	Required	Description
zoneID	integer	no	Unique id allocated by IAS CIE
numberofzonesensitivitylevelsupported	integer	no	Total number of sensitivity levels supported by the IAS Zone
zonestate	boolean	yes	Enrollment status of IAS Zone false=not enrolled, true=enrolled
IAS_CIE_address	string	no	Address of IAS Control and Indicating Equipment (CIE)
zonetype	string	no	Zonetype and Meaning of Alarm1 and Alarm2 zonestatus
zonestatus	array	no	x is a variable. zonestatus in Zigbee maps to zonestatus, zonebattery, and zonepowersource in OCF. Data type of zonestatus in Zigbee is 16 bitmap (xxxxxxxxxxxxxxxx) : bit 0 = Alarm1, bit 1 = Alarm2, bit 2 = Tamper, bit 3 = Battery, bit 4 = Supervision reports, bit 5 = Restore reports, bit 6 = Trouble, bit 7 = AC (mains), bit 8 = Test, bit 9 = Battery Defect. Alarm1 : 1 = opened or alarmed 0 = closed or not alarmed, Alarm2 : 1 = opened or alarmed 0 = closed or not alarmed, Tamper : 1 = Tampered 0 = Not tampered, Battery : 1 = Low battery 0 = Battery OK, Supervision reports : 1 = Reports 0 = Does not report, Restore reports : 1 = Reports restore 0 = Does not

			report restore, Trouble : 1 = Trouble/Failure 0 = OK, AC (mains) : 1 = AC/Mains fault 0 = AC/Mains OK, Test : 1 = Sensor is in test mode 0 = Sensor is in operation mode, Battery Defect : 1 = Sensor detects a defective battery 0 = Sensor battery is functioning.
currentzonesensitivitylevel	integer	no	Sensitivity level of IAS Zone

1039 **9.10.3 Derived model definition**

```

1040 {
1041   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.iaszone.info.json#",
1042   "$schema": "http://json-schema.org/draft-04/schema#",
1043   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1044   "title": "IAS Zone Cluster - Information",
1045   "definitions": {
1046     "zcl.iaszone.info": {
1047       "type": "object",
1048       "properties": {
1049         "zonestate": {
1050           "type": "boolean",
1051           "description": "Enrollment status of IAS Zone false=not enrolled, true=enrolled",
1052           "x-ocf-conversion": {
1053             "x-ocf-alias": "oic.r.iaszone",
1054             "x-to-ocf": [
1055               "if zonestate=0x00, ocf.zonestate=false",
1056               "if zonestate=0x01, ocf.zonestate=true"
1057             ],
1058             "x-from-ocf": [
1059               "N/A"
1060             ]
1061           }
1062         },
1063         "zonetype": {
1064           "type": "string",
1065           "description": "Zonetype and Meaning of Alarm1 and Alarm2 zonestatus",
1066           "x-ocf-conversion": {
1067             "x-ocf-alias": "oic.r.iaszone",
1068             "x-to-ocf": [
1069               "if zonetype=0x0000, ocf.zonetype=Standard CIE",
1070               "if zonetype=0x000d, ocf.zonetype=Motion sensor",
1071               "if zonetype=0x0015, ocf.zonetype=Contact switch",
1072               "if zonetype=0x0028, ocf.zonetype=Fire sensor",
1073               "if zonetype=0x002a, ocf.zonetype=Water sensor",
1074               "if zonetype=0x002b, ocf.zonetype=Carbon Monoxide (CO) sensor",
1075               "if zonetype=0x002c, ocf.zonetype=Personal emergency device",
1076               "if zonetype=0x002d, ocf.zonetype=Vibration/Movement sensor",
1077               "if zonetype=0x010f, ocf.zonetype=Remote Control",
1078               "if zonetype=0x0115, ocf.zonetype=Key fob",
1079               "if zonetype=0x021d, ocf.zonetype=Keypad",
1080               "if zonetype=0x0225, ocf.zonetype=Standard Warning Device",
1081               "if zonetype=0x0226, ocf.zonetype=Glass break sensor",
1082               "if zonetype=0x0229, ocf.zonetype=Security repeater",
1083               "if zonetype=0xffff, ocf.zonetype=Invalid Zone Type"
1084             ],
1085             "x-from-ocf": [
1086               "N/A"
1087             ]
1088           }
1089         }
1090       }
1091     }
1092   }

```

```

1088     }
1089   },
1090   "zonestatus": {
1091     "type": "array",
1092     "items": {
1093       "type": "integer"
1094     },
1095     "description": "x is a variable. zonestatus in Zigbee maps to zonestatus, zonebattery,
1096 and zonepowersource in OCF. Data type of zonestatus in Zigbee is 16 bitmap (xxxxxxxxxxxxxxxx) : bit
1097 0 = Alarm1, bit 1 = Alarm2, bit 2 = Tamper, bit 3 = Battery, bit 4 = Supervision reports, bit 5 =
1098 Restore reports, bit 6 = Trouble, bit 7 = AC (mains), bit 8 = Test, bit 9 = Battery Defect.
1099 Alarm1 : 1 = opened or alarmed 0 = closed or not alarmed, Alarm2 : 1 = opened or alarmed 0 = closed
1100 or not alarmed, Tamper : 1 = Tampered 0 = Not tampered, Battery : 1 = Low battery 0 = Battery OK,
1101 Supervision reports : 1 = Reports 0 = Does not report, Restore reports : 1 = Reports restore 0 =
1102 Does not report restore, Trouble : 1 = Trouble/Failure 0 = OK, AC (mains) : 1 = AC/Mains fault 0 =
1103 AC/Mains OK, Test : 1 = Sensor is in test mode 0 = Sensor is in operation mode, Battery Defect : 1
1104 = Sensor detects a defective battery 0 = Sensor battery is functioning.",
1105     "x-ocf-conversion": {
1106       "x-ocf-alias": "oic.r.iaszone",
1107       "x-to-ocf": [
1108
1109         "if zonetype=0x0000 & zonestatus=xxxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
1110         "if zonetype=0x0000 & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['system']",
1111         "if zonetype=0x0000 & zonestatus=xxxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
1112         "if zonetype=0x0000 & zonestatus=xxxxxxxxxxxxx1x, ocf.zonestatus.alarms=['']",
1113
1114         "if zonetype=0x000d & zonestatus=xxxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
1115         "if zonetype=0x000d & zonestatus=xxxxxxxxxxxxx1,
1116 ocf.zonestatus.alarms=['intrusion']",
1117         "if zonetype=0x000d & zonestatus=xxxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
1118         "if zonetype=0x000d & zonestatus=xxxxxxxxxxxxx1x,
1119 ocf.zonestatus.alarms=['presence']",
1120         "if zonetype=0x000d & zonestatus=xxxxxxxxxxxxx11,
1121 ocf.zonestatus.alarms=['intrusion','presence']",
1122
1123         "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
1124         "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxxx1,
1125 ocf.zonestatus.alarms=['1stportalopenclose']",
1126         "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
1127         "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxxx1x,
1128 ocf.zonestatus.alarms=['2ndportalopenclose']",
1129         "if zonetype=0x0015 & zonestatus=xxxxxxxxxxxxx11,
1130 ocf.zonestatus.alarms=['1stportalopenclose','2ndportalopenclose']",
1131
1132         "if zonetype=0x0028 & zonestatus=xxxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
1133         "if zonetype=0x0028 & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['fire']",
1134         "if zonetype=0x0028 & zonestatus=xxxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
1135         "if zonetype=0x0028 & zonestatus=xxxxxxxxxxxxx1x, ocf.zonestatus.alarms=['']",
1136
1137         "if zonetype=0x002a & zonestatus=xxxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
1138         "if zonetype=0x002a & zonestatus=xxxxxxxxxxxxx1,
1139 ocf.zonestatus.alarms=['wateroverflow']",
1140         "if zonetype=0x002a & zonestatus=xxxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
1141         "if zonetype=0x002a & zonestatus=xxxxxxxxxxxxx1x, ocf.zonestatus.alarms=['']",
1142
1143         "if zonetype=0x002b & zonestatus=xxxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
1144         "if zonetype=0x002b & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['CO']",
1145         "if zonetype=0x002b & zonestatus=xxxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
1146         "if zonetype=0x002b & zonestatus=xxxxxxxxxxxxx1x,
1147 ocf.zonestatus.alarms=['cooking']",
1148         "if zonetype=0x002b & zonestatus=xxxxxxxxxxxxx11,
1149 ocf.zonestatus.alarms=['CO','cooking']",
1150
1151         "if zonetype=0x002c & zonestatus=xxxxxxxxxxxxx0, ocf.zonestatus.alarms=['']",
1152         "if zonetype=0x002c & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['fall']",
1153         "if zonetype=0x002c & zonestatus=xxxxxxxxxxxxx0x, ocf.zonestatus.alarms=['']",
1154         "if zonetype=0x002c & zonestatus=xxxxxxxxxxxxx1x,
1155 ocf.zonestatus.alarms=['emergencybutton']",
1156         "if zonetype=0x002c & zonestatus=xxxxxxxxxxxxx11,
1157 ocf.zonestatus.alarms=['fall','emergencybutton']",
1158

```

```

1159         "if zonetype=0x002d & zonestatus=xxxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1160         "if zonetype=0x002d & zonestatus=xxxxxxxxxxxxx1,
1161 ocf.zonestatus.alarms=['movement']",
1162         "if zonetype=0x002d & zonestatus=xxxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1163         "if zonetype=0x002d & zonestatus=xxxxxxxxxxxxx1x,
1164 ocf.zonestatus.alarms=['vibration']",
1165         "if zonetype=0x002d & zonestatus=xxxxxxxxxxxxx11,
1166 ocf.zonestatus.alarms=['movement', 'vibration']",
1167
1168         "if zonetype=0x010f & zonestatus=xxxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1169         "if zonetype=0x010f & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['panic']",
1170         "if zonetype=0x010f & zonestatus=xxxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1171         "if zonetype=0x010f & zonestatus=xxxxxxxxxxxxx1x,
1172 ocf.zonestatus.alarms=['emergency']",
1173         "if zonetype=0x010f & zonestatus=xxxxxxxxxxxxx11,
1174 ocf.zonestatus.alarms=['panic', 'emergency']",
1175
1176         "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1177         "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['panic']",
1178         "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1179         "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxxx1x,
1180 ocf.zonestatus.alarms=['emergency']",
1181         "if zonetype=0x0115 & zonestatus=xxxxxxxxxxxxx11,
1182 ocf.zonestatus.alarms=['panic', 'emergency']",
1183
1184         "if zonetype=0x021d & zonestatus=xxxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1185         "if zonetype=0x021d & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=['panic']",
1186         "if zonetype=0x021d & zonestatus=xxxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1187         "if zonetype=0x021d & zonestatus=xxxxxxxxxxxxx1x,
1188 ocf.zonestatus.alarms=['emergency']",
1189         "if zonetype=0x021d & zonestatus=xxxxxxxxxxxxx11,
1190 ocf.zonestatus.alarms=['panic', 'emergency']",
1191
1192         "if zonetype=0x0225 & zonestatus=xxxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1193         "if zonetype=0x0225 & zonestatus=xxxxxxxxxxxxx1,
1194 ocf.zonestatus.alarms=['glassbreak']",
1195         "if zonetype=0x0225 & zonestatus=xxxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1196         "if zonetype=0x0225 & zonestatus=xxxxxxxxxxxxx1x, ocf.zonestatus.alarms=[' ']",
1197
1198         "if zonetype=0x0226 & zonestatus=xxxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1199         "if zonetype=0x0226 & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=[' ']",
1200         "if zonetype=0x0226 & zonestatus=xxxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1201         "if zonetype=0x0226 & zonestatus=xxxxxxxxxxxxx1x, ocf.zonestatus.alarms=[' ']",
1202
1203         "if zonetype=0x0229 & zonestatus=xxxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1204         "if zonetype=0x0229 & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=[' ']",
1205         "if zonetype=0x0229 & zonestatus=xxxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1206         "if zonetype=0x0229 & zonestatus=xxxxxxxxxxxxx1x, ocf.zonestatus.alarms=[' ']",
1207
1208         "if zonetype=0xffff & zonestatus=xxxxxxxxxxxxx0, ocf.zonestatus.alarms=[' ']",
1209         "if zonetype=0xffff & zonestatus=xxxxxxxxxxxxx1, ocf.zonestatus.alarms=[' ']",
1210         "if zonetype=0xffff & zonestatus=xxxxxxxxxxxxx0x, ocf.zonestatus.alarms=[' ']",
1211         "if zonetype=0xffff & zonestatus=xxxxxxxxxxxxx1x, ocf.zonestatus.alarms=[' ']",
1212
1213         "if zonestatus=xxxxxxxxxxxxx0xx, ocf.zonestatus.tamper=false",
1214         "if zonestatus=xxxxxxxxxxxxx1xx, ocf.zonestatus.tamper=true",
1215
1216         "if zonestatus=xxxxxxxxxxxxx0xxx, ocf.zonebattery.charge=100 &
1217 ocf.zonebattery.lowbattery=false",
1218         "if zonestatus=xxxxxxxxxxxxx1xxx, ocf.zonebattery.charge=100 &
1219 ocf.zonebattery.lowbattery=true",
1220
1221         "if zonestatus=xxxxxxxx00xxxx, ocf.zonestatus.zonestatusreports='none'",
1222         "if zonestatus=xxxxxxxx01xxxx, ocf.zonestatus.zonestatusreports='statuschangeonly'
1223 ",
1224         "if zonestatus=xxxxxxxx10xxxx, ocf.zonestatus.zonestatusreports='alarmclearonly'",
1225         "if zonestatus=xxxxxxxx11xxxx,
1226 ocf.zonestatus.zonestatusreports='statuschangeandalarmclear'",
1227
1228         "if zonestatus=xxxxxxxx0xxxxx, ocf.zonestatus.fault=false",
1229         "if zonestatus=xxxxxxxx1xxxxx, ocf.zonestatus.fault=true",

```

```

1230
1231         "if zonestatus=xxxxxx0xxxxxxx, ocf.zonepowersource.powerSources=['AC (Mains)
1232 Power'] & ocf.zonepowersource.sourcedefault=false",
1233         "if zonestatus=xxxxxxlxxxxxxx, ocf.zonepowersource.powerSources=['AC (Mains)
1234 Power'] & ocf.zonepowersource.sourcedefault=true",
1235
1236         "if zonestatus=xxxxx0xxxxxxx, ocf.zonestatus.test=false",
1237         "if zonestatus=xxxxxlxxxxxxx, ocf.zonestatus.test=true",
1238
1239         "if zonestatus=xxxx0xxxxxxx, ocf.zonepowersource.powerSources=['Internal
1240 Battery'] & oic.r.ias.zone.zonebattery.defect=false & oic.r.ias.zone.zonebattery.charge=100.",
1241         "if zonestatus=xxxxlxxxxxxx,
1242 oic.r.ias.zone.zonepowersource.powerSources=['Internal Battery'] &
1243 oic.r.ias.zone.zonebattery.defect=true & oic.r.ias.zone.zonebattery.charge=100."
1244     ],
1245     "x-from-ocf": [
1246         "N/A"
1247     ]
1248 },
1249 },
1250 "IAS_CIE_address": {
1251     "type": "string",
1252     "description": "Address of IAS Control and Indicating Equipment (CIE)",
1253     "x-ocf-conversion": {
1254         "x-ocf-alias": "oic.r.iaszone",
1255         "x-to-ocf": [
1256             "ocf.iascieaddress= IAS_CIE_address"
1257         ],
1258         "x-from-ocf": [
1259             "N/A"
1260         ]
1261     }
1262 },
1263 "zoneID": {
1264     "type": "integer",
1265     "description": "Unique id allocated by IAS CIE",
1266     "x-ocf-conversion": {
1267         "x-ocf-alias": "oic.r.iaszone",
1268         "x-to-ocf": [
1269             "ocf.zoneid=zoneID"
1270         ],
1271         "x-from-ocf": [
1272             "N/A"
1273         ]
1274     }
1275 },
1276 "numberofzonesensitivitylevelsupported": {
1277     "type": "integer",
1278     "description": "Total number of sensitivity levels supported by the IAS Zone",
1279     "x-ocf-conversion": {
1280         "x-ocf-alias": "oic.r.iaszone",
1281         "x-to-ocf": [
1282             "ocf.numzonesensitivitylevel= numberofzonesensitivitylevelsupported"
1283         ],
1284         "x-from-ocf": [
1285             "N/A"
1286         ]
1287     }
1288 },
1289 "currentzonesensitivitylevel": {
1290     "type": "integer",
1291     "description": "Sensitivity level of IAS Zone",
1292     "x-ocf-conversion": {
1293         "x-ocf-alias": "oic.r.iaszone",
1294         "x-to-ocf": [
1295             "ocf.currentzonesensitivitylevel = currentzonesensitivitylevel"
1296         ],
1297         "x-from-ocf": [
1298             "N/A"
1299         ]
1300     }

```

```

1301     }
1302   }
1303 }
1304 },
1305 "type": "object",
1306 "allOf": [
1307   {"$ref": "#/definitions/zcl.iaszone.info"}
1308 ],
1309 "required": [ "zonestate" ]
1310 }
1311

```

9.11 Level Control Cluster - Control

9.11.1 Derived model

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

9.11.2 Property definition

Table 25 provides the detailed per Property mapping for "zcl.levelcontrol.control.moveto".

Table 25 – The Property mapping for "zcl.levelcontrol.control.moveto".

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

Table 26 provides the details of the Properties that are part of "zcl.levelcontrol.control.moveto".

Table 26 – 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

9.11.3 Derived model definition

```

1321 {
1322   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.levelcontrol.control.json#",
1323   "$schema": "http://json-schema.org/draft-04/schema#",
1324   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1325   "title": "Level Control Cluster - Control",
1326   "definitions": {
1327     "zcl.levelcontrol.control.moveto": {
1328       "properties": {
1329         "level": {
1330           "type": "integer",
1331           "description": "Move to certain dimming value as fast as possible ",
1332           "x-ocf-conversion": {
1333             "x-ocf-alias": "oic.r.light.dimming",
1334             "x-from-ocf": [
1335               "level=ocf.dimmingSetting * 254 /100 , transitiontime=0",
1336               "zcl.command.levelcontrol::movetolevel(level,transitiontime)"
1337             ],
1338             "x-to-ocf": [
1339               "N/A"
1340             ]
1341           }
1342         }
1343       }
1344     }

```

```

1345     },
1346     "type": "object",
1347     "allOf": [
1348       { "$ref": "#/definitions/zcl.levelcontrol.control.movetolevel" }
1349     ]
1350   }
1351 }

```

1352 9.12 Level Control Cluster - Information

1353 9.12.1 Derived model

1354 The derived model: "zcl.levelcontrol.info".

1355 9.12.2 Property definition

1356 Table 27 provides the detailed per Property mapping for "zcl.levelcontrol.info".

1357 **Table 27 – The Property mapping for "zcl.levelcontrol.info".**

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

1358 Table 28 provides the details of the Properties that are part of "zcl.levelcontrol.info".

1359 **Table 28 – The Properties of "zcl.levelcontrol.info".**

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

1360 9.12.3 Derived model definition

```

1361 {
1362   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.levelcontrol.info.json#",
1363   "$schema": "http://json-schema.org/draft-04/schema#",
1364   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1365   "title": "Level Control Cluster - Information",
1366   "definitions": {
1367     "zcl.levelcontrol.info": {
1368       "type": "object",
1369       "properties": {
1370         "currentlevel": {
1371           "type": "integer",
1372           "description": "current dimming value",
1373           "x-ocf-conversion": {
1374             "x-ocf-alias": "oic.r.light.dimming",
1375             "x-to-ocf": [
1376               "ocf.dimmingsetting = currentlevel/254 * 100"
1377             ],
1378             "x-from-ocf": [
1379               "N/A"
1380             ]
1381           }
1382         }
1383       }
1384     }
1385   },
1386   "type": "object",
1387   "allOf": [
1388     { "$ref": "#/definitions/zcl.levelcontrol.info" }
1389   ],
1390   "required": [ "currentlevel" ]
1391 }
1392 }

```


1393 **9.13 Occupancy Sensing Cluster - Information**

1394 **9.13.1 Derived model**

1395 The derived model: "zcl.occupancysensing.info".

1396 **9.13.2 Property definition**

1397 Table 29 provides the detailed per Property mapping for "zcl.occupancysensing.info".

1398 **Table 29 – The Property mapping for "zcl.occupancysensing.info".**

Zigbee name	Property	OCF Resource	To OCF	From OCF
occupancy		oic.r.sensor.presence	if occupancy =xxxxxxx0, then ocf.value = false if occupancy =xxxxxxx1, then ocf.value = true	N/A

1399 Table 30 provides the details of the Properties that are part of "zcl.occupancysensing.info".

1400 **Table 30 – 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

1401 **9.13.3 Derived model definition**

```

1402 {
1403   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.occupancysensing.info.json#",
1404   "$schema": "http://json-schema.org/draft-04/schema#",
1405   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1406   "title": "Occupancy Sensing Cluster - Information",
1407   "definitions": {
1408     "zcl.occupancysensing.info": {
1409       "type": "object",
1410       "properties": {
1411         "occupancy": {
1412           "type": "number",
1413           "description": "x is a variable. Data type of occupancy in Zigbee is 8 bitmap (xxxxxxx)
1414 while data type of value in OCF is boolean type i.e., true=occupied, false=unoccupied",
1415           "x-ocf-conversion": {
1416             "x-ocf-alias": "oic.r.sensor.presence",
1417             "x-to-ocf": [
1418               "if occupancy =xxxxxxx0, then ocf.value = false",
1419               "if occupancy =xxxxxxx1, then ocf.value = true"
1420             ],
1421             "x-from-ocf": [
1422               "N/A"
1423             ]
1424           }
1425         }
1426       }
1427     },
1428   },
1429   "type": "object",

```

```

1430     "allOf": [
1431       {"$ref": "#/definitions/zcl.occupancysensing.info"}
1432     ],
1433     "required": [ "occupancy" ]
1434   }
1435

```

1436 9.14 On/Off Cluster - Control

1437 9.14.1 Derived model

1438 The derived model: "zcl.onoff.control.off".

1439 The derived model: "zcl.onoff.control.on".

1440 9.14.2 Property definition

1441 Table 31 provides the detailed per Property mapping for "zcl.onoff.control.off".

1442 **Table 31 – 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().

1443 Table 32 provides the details of the Properties that are part of "zcl.onoff.control.off".

1444 **Table 32 – The Properties of "zcl.onoff.control.off".**

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

1445 Table 33 provides the detailed per Property mapping for "zcl.onoff.control.on".

1446 **Table 33 – 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().

1447 Table 34 provides the details of the Properties that are part of "zcl.onoff.control.on".

1448 **Table 34 – The Properties of "zcl.onoff.control.on".**

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

1449 9.14.3 Derived model definition

```

1450 {
1451   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.onoff.control.json#",
1452   "$schema": "http://json-schema.org/draft-04/schema#",
1453   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1454   "title": "On/Off Cluster - Control",
1455   "definitions": {
1456     "zcl.onoff.control.on": {
1457       "properties": {
1458         "onoff": {
1459           "type": "boolean",
1460           "description": "Turn on the device",
1461           "x-ocf-conversion": {
1462             "x-ocf-alias": "oic.r.switch.binary",

```

```

1463         "x-from-ocf": [
1464             "if ocf.value = true, zcl.command.onoff::on()."
1465         ],
1466         "x-to-ocf": [
1467             "N/A"
1468         ]
1469     }
1470 }
1471 }
1472 },
1473 "zcl.onoff.control.off": {
1474     "properties": {
1475         "onoff": {
1476             "type": "boolean",
1477             "description": "Turn off the device",
1478             "x-ocf-conversion": {
1479                 "x-ocf-alias": "oic.r.switch.binary",
1480                 "x-from-ocf": [
1481                     "if ocf.value = false, zcl.command.onoff::off()."
1482                 ],
1483                 "x-to-ocf": [
1484                     "N/A"
1485                 ]
1486             }
1487         }
1488     }
1489 },
1490 "type": "object",
1491 "allOf": [
1492     {"$ref": "#/definitions/zcl.onoff.control.on"},
1493     {"$ref": "#/definitions/zcl.onoff.control.off"}
1494 ]
1495 }
1496 }
1497

```

1498 9.15 On/off Cluster - Information

1499 9.15.1 Derived model

1500 The derived model: "zcl.onoff".

1501 9.15.2 Property definition

1502 Table 35 provides the detailed per Property mapping for "zcl.onoff".

1503 **Table 35 – 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 = falseif onoff = true, then ocf.value = true	N/A

1504 Table 36 provides the details of the Properties that are part of "zcl.onoff".

1505 **Table 36 – The Properties of "zcl.onoff".**

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

1506 9.15.3 Derived model definition

```

1507 {
1508     "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.onoff.info.json#",

```

```

1509 "$schema": "http://json-schema.org/draft-04/schema#",
1510 "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1511 "title": "On/off Cluster - Information",
1512 "definitions": {
1513   "zcl.onoff": {
1514     "type": "object",
1515     "properties": {
1516       "onoff": {
1517         "type": "boolean",
1518         "description": "On/off status of the device",
1519         "x-ocf-conversion": {
1520           "x-ocf-alias": "oic.r.switch.binary",
1521           "x-to-ocf": [
1522             "if onoff = false, then ocf.value = false",
1523             "if onoff = true, then ocf.value = true"
1524           ],
1525           "x-from-ocf": [
1526             "N/A"
1527           ]
1528         }
1529       }
1530     }
1531   },
1532 },
1533 "type": "object",
1534 "allOf": [
1535   {"$ref": "#/definitions/zcl.onoff.info"}
1536 ],
1537 "required": [ "onoff" ]
1538 }
1539

```

1540 9.16 Temperature Measurement Cluster - Information

1541 9.16.1 Derived model

1542 The derived model: "zcl.temperaturemeasurement.info".

1543 9.16.2 Property definition

1544 Table 37 provides the detailed per Property mapping for "zcl.temperaturemeasurement.info".

1545 **Table 37 – The Property mapping for "zcl.temperaturemeasurement.info".**

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

1546 Table 38 provides the details of the Properties that are part of "zcl.temperaturemeasurement.info".

1547 **Table 38 – The Properties of "zcl.temperaturemeasurement.info".**

Zigbee Property name	Type	Required	Description
MeasuredValue	number	yes	Measured value
MinMeasuredValue	number	yes	Minimum value of MeasuredValue

Tolerance	number	yes	Magnitude of the possible error
MaxMeasuredValue	number	yes	Maximum value of MeasuredValue

1548 **9.16.3 Derived model definition**

```

1549 {
1550   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.temperaturemeasurement.info.json#",
1551   "$schema": "http://json-schema.org/draft-04/schema#",
1552   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1553   "title": "Temperature Measurement Cluster - Information",
1554   "definitions": {
1555     "zcl.temperaturemeasurement.info": {
1556       "type": "object",
1557       "properties": {
1558         "MeasuredValue": {
1559           "type": "number",
1560           "description": "Measured value",
1561           "x-ocf-conversion": {
1562             "x-ocf-alias": "oic.r.temperature",
1563             "x-to-ocf": [
1564               "ocf.temperature = MeasuredValue/100",
1565               "units = C"
1566             ],
1567             "x-from-ocf": [
1568               "N/A"
1569             ]
1570           }
1571         },
1572         "Tolerance": {
1573           "type": "number",
1574           "description": "Magnitude of the possible error",
1575           "x-ocf-conversion": {
1576             "x-ocf-alias": "oic.r.temperature",
1577             "x-to-ocf": [
1578               "ocf.precision = Tolerance/100"
1579             ],
1580             "x-from-ocf": [
1581               "N/A"
1582             ]
1583           }
1584         },
1585         "MinMeasuredValue": {
1586           "type": "number",
1587           "description": "Minimum value of MeasuredValue",
1588           "x-ocf-conversion": {
1589             "x-ocf-alias": "oic.r.temperature",
1590             "x-to-ocf": [
1591               "ocf.range[0] = MinMeasuredValue/100"
1592             ],
1593             "x-from-ocf": [
1594               "N/A"
1595             ]
1596         },
1597         "MaxMeasuredValue": {
1598           "type": "number",
1599           "description": "Maximum value of MeasuredValue",
1600           "x-ocf-conversion": {
1601             "x-ocf-alias": "oic.r.temperature",
1602             "x-to-ocf": [
1603               "ocf.range[1] = MaxMeasuredValue/100"
1604             ],
1605             "x-from-ocf": [
1606               "N/A"
1607             ]
1608         }
1609       }
1610     }
1611   }

```

```

1612     }
1613   },
1614   "type": "object",
1615   "allOf": [
1616     { "$ref": "#/definitions/zcl.temperaturemeasurement.info" }
1617   ],
1618   "required": [ "MeasuredValue", "Tolerance", "MinMeasuredValue", "MaxMeasuredValue" ]
1619 }
1620

```

1621 9.17 Thermostat Cluster - Cool - Control

1622 9.17.1 Derived model

1623 The derived model: "zcl.thermostat_cool.control.setpointraiselower".

1624 9.17.2 Property definition

1625 Table 39 provides the detailed per Property mapping for
 1626 "zcl.thermostat_cool.control.setpointraiselower".

1627 **Table 39 – 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)

1628 Table 40 provides the details of the Properties that are part of
 1629 "zcl.thermostat_cool.control.setpointraiselower".

1630 **Table 40 – 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

1631 9.17.3 Derived model definition

```

1632 {
1633   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.thermostat_cool.control.json#",
1634   "$schema": "http://json-schema.org/draft-04/schema#",
1635   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1636   "title": "Thermostat Cluster - Cool - Control",
1637   "definitions": {
1638     "zcl.thermostat_cool.control.setpointraiselower": {
1639       "type": "object",
1640       "properties": {
1641         "amount": {
1642           "type": "number",
1643           "description": "Set the target temperature with cool mode. Mode=0x01 is set by Zigbee 3.0
1644 translator",
1645           "x-ocf-conversion": {
1646             "x-ocf-alias": "oic.r.temperature",
1647             "x-from-ocf": [
1648               "if ocf.temperature is updated, then amount= ocf.temperature*100.",
1649               "zcl.command.thermostat::setpointraiselower(mode, amount)"
1650             ],
1651             "x-to-ocf": [
1652               "N/A"

```

```

1653     ]
1654   }
1655 }
1656 }
1657 },
1658 },
1659 "type": "object",
1660 "allOf": [
1661   {"$ref": "#/definitions/zcl.thermostat_cool.control.setpointraiselower"}
1662 ]
1663 }
1664

```

1665 9.18 Thermostat Cluster - Current Temperature - Information

1666 9.18.1 Derived model

1667 The derived model: "zcl.thermostat_currenttemperature.info".

1668 9.18.2 Property definition

1669 Table 41 provides the detailed per Property mapping for "zcl.thermostat_currenttemperature.info".

1670 **Table 41 – The Property mapping for "zcl.thermostat_currenttemperature.info".**

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

1671 Table 42 provides the details of the Properties that are part of
1672 "zcl.thermostat_currenttemperature.info".

1673 **Table 42 – The Properties of "zcl.thermostat_currenttemperature.info".**

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

1674 9.18.3 Derived model definition

```

1675 {
1676   "id":
1677   "http://openinterconnect.org/zigbeemapping/schemas/zcl.thermostat_currenttemperature.info.json#",
1678   "$schema": "http://json-schema.org/draft-04/schema#",
1679   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1680   "title": "Thermostat Cluster - Current Temperature - Information ",
1681   "definitions": {
1682     "zcl.thermostat_currenttemperature.info": {
1683       "type": "object",
1684       "properties": {
1685         "localtemperature": {
1686           "type": "number",
1687           "description": "current sensed temperature",
1688           "x-ocf-conversion": {
1689             "x-ocf-alias": "oic.r.temperature",
1690             "x-to-ocf": [
1691               "ocf.temperature=localtempearture/100",
1692               "units = C"
1693             ],
1694             "x-from-ocf": [
1695               "N/A"
1696             ]
1697           }
1698         }
1699       }
1700     }

```

```

1701     },
1702     "type": "object",
1703     "allOf": [
1704       {"$ref": "#/definitions/zcl.thermostat_currenttemperature.info"}
1705     ],
1706     "required": [ "localtempearture" ]
1707   }
1708 }

```

1709 **9.19 Thermostat Cluster - Heat - Control**

1710 **9.19.1 Derived model**

1711 The derived model: "zcl.thermostat_heat.control.setpointraiseLower".

1712 **9.19.2 Property definition**

1713 Table 43 provides the detailed per Property mapping for
 1714 "zcl.thermostat_heat.control.setpointraiseLower".

1715 **Table 43 – 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)

1716 Table 44 provides the details of the Properties that are part of
 1717 "zcl.thermostat_heat.control.setpointraiseLower".

1718 **Table 44 – 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

1719 **9.19.3 Derived model definition**

```

1720 {
1721   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.thermostat_heat.control.json#",
1722   "$schema": "http://json-schema.org/draft-04/schema#",
1723   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1724   "title": "Thermostat Cluster - Heat - Control",
1725   "definitions": {
1726     "zcl.thermostat_heat.control.setpointraiseLower": {
1727       "type": "object",
1728       "properties": {
1729         "amount": {
1730           "type": "number",
1731           "description": "Set the target temperature with heat mode. Mode=0x00 is set by Zigbee 3.0
1732 translator",
1733           "x-ocf-conversion": {
1734             "x-ocf-alias": "oic.r.temperature",
1735             "x-from-ocf": [
1736               "if ocf.temperature is updated, then amount= ocf.temperature*100.",
1737               "zcl.command.thermostat::setpointraiseLower(mode, amount)"
1738             ],
1739             "x-to-ocf": [
1740               "N/A"
1741             ]
1742           }
1743         }
1744       }
1745     }
1746   }

```



```

1742     }
1743   }
1744 }
1745 }
1746 },
1747 "type": "object",
1748 "allOf": [
1749   {"$ref": "#/definitions/zcl.thermostat_heat.control.setpointraiseLower"}
1750 ]
1751 }
1752

```

1753 **9.20 Window Covering Cluster - Configuration - Control**

1754 **9.20.1 Derived model**

1755 The derived model: "zcl.windowcovering_conf.control".

1756 **9.20.2 Property definition**

1757 Table 45 provides the detailed per Property mapping for "zcl.windowcovering_conf.control".

1758 **Table 45 – The Property mapping for "zcl.windowcovering_conf.control".**

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

1759 Table 46 provides the details of the Properties that are part of "zcl.windowcovering_conf.control".

1760 **Table 46 – The Properties of "zcl.windowcovering_conf.control".**

Zigbee name	Property	Type	Required	Description
	Acceleration Time-Lift	integer	no	Set ramp up times to reaching the velocity setting (0.1sec).
	Velocity-Lift	integer	no	Set velocity associated with Lifting the Window Covering (cm/sec).

Deceleration Time-Lift	integer	no	Set ramp down times associated with stopping the velocity setting (0.1sec).
mode	integer	no	Set the mode. x is a variable. Data type of Mode in Zigbee is 8 bitmap (xxxxxxx) while data type of mode in OCF is array with 4 Boolean type items(i.e., [Reversed Motor Direction, Calibration Mode, Maintenance Mode, LED]). Reversed Motor Direction : 0 = motor direction is normal, 1 = motor direction is reversed. Calibration Mode : 0 = run in normal mode, 1 = run in calibration mode. Maintenance Mode : 0 = motor is running normally, 1 = motor is running in maintenance mode. LED: 0 = LEDs are off, 1 = LEDs will display feedback.

1761 **9.20.3 Derived model definition**

```

1762 {
1763   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_conf.control.json#",
1764   "$schema": "http://json-schema.org/draft-04/schema#",
1765   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1766   "title": "Window Covering Cluster - Configuration - Control",
1767   "definitions": {
1768     "zcl.windowcovering_conf.control": {
1769       "properties": {
1770         "mode": {
1771           "type": "integer",
1772           "description": "Set the mode. x is a variable. Data type of Mode in Zigbee is 8 bitmap
1773 (xxxxxxx) while data type of mode in OCF is array with 4 Boolean type items(i.e., [Reversed Motor
1774 Direction, Calibration Mode, Maintenance Mode, LED]). Reversed Motor Direction : 0 = motor
1775 direction is normal, 1 = motor direction is reversed. Calibration Mode : 0 = run in normal mode, 1
1776 = run in calibration mode. Maintenance Mode : 0 = motor is running normally, 1 = motor is running
1777 in maintenance mode. LED: 0 = LEDs are off, 1 = LEDs will display feedback.",
1778         "x-ocf-conversion": {
1779           "x-ocf-alias": "oic.r.windowcovering",
1780           "x-from-ocf": [
1781             "if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxxxxx0.",
1782             "if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxxxxx1.",
1783             "if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxxxxx0x.",
1784             "if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxxxxx1x.",
1785             "if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxxx0xxx.",
1786             "if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxxx1xxx.",
1787             "if ocf.mode is updated & ocf.mode = [false,x,x,x], Mode =xxxx0xxx.",
1788             "if ocf.mode is updated & ocf.mode = [true,x,x,x], Mode =xxxx1xxx.",
1789             "zcl.command.general::write(mode)"
1790           ],
1791         }
1792       }
1793     }
1794   }

```

```

1791         "x-to-ocf": [
1792             "N/A"
1793         ]
1794     },
1795 },
1796 "Velocity-Lift": {
1797     "type": "integer",
1798     "description": "Set velocity associated with Lifting the Window Covering (cm/sec).",
1799     "x-ocf-conversion": {
1800         "x-ocf-alias": "oic.r.windowcovering",
1801         "x-from-ocf": [
1802             "if ocf.liftvelocity is updated, Velocity-Lift = ocf.liftvelocity.",
1803             "zcl.command.general::write(Velocity-Lift)"
1804         ],
1805         "x-to-ocf": [
1806             "N/A"
1807         ]
1808     }
1809 },
1810 "Acceleration Time-Lift": {
1811     "type": "integer",
1812     "description": "Set ramp up times to reaching the velocity setting (0.1sec).",
1813     "x-ocf-conversion": {
1814         "x-ocf-alias": "oic.r.windowcovering",
1815         "x-from-ocf": [
1816             "if ocf.liftaccelerationtime is updated, Acceleration Time-
1817 Lift=ocf.liftaccelerationtime.",
1818             "zcl.command.general::write(Acceleration Time-Lift)"
1819         ],
1820         "x-to-ocf": [
1821             "N/A"
1822         ]
1823     }
1824 },
1825 "Deceleration Time-Lift": {
1826     "type": "integer",
1827     "description": "Set ramp down times associated with stoping the velocity setting
1828 (0.1sec).",
1829     "x-ocf-conversion": {
1830         "x-ocf-alias": "oic.r.windowcovering",
1831         "x-from-ocf": [
1832             "if ocf.liftdecelerationtime is updated, Deceleration Time-
1833 Lift=ocf.liftdecelerationtime.",
1834             "zcl.command.general::write(Deceleration Time-Lift)"
1835         ],
1836         "x-to-ocf": [
1837             "N/A"
1838         ]
1839     }
1840 }
1841 }
1842 }
1843 },
1844 "type": "object",
1845 "allOf": [
1846     {"$ref": "#/definitions/zcl.windowcovering_conf.control"}
1847 ]
1848 }
1849

```

1850 9.21 Window Covering Cluster - Configuration - Information

1851 9.21.1 Derived model

1852 The derived model: "zcl.windowcovering_conf.info".

1853 9.21.2 Property definition

1854 Table 47 provides the detailed per Property mapping for "zcl.windowcovering_conf.info".

Table 47 – The Property mapping for "zcl.windowcovering_conf.info".

Zigbee name	Property	OCF Resource	To OCF	From OCF
Velocity-Lift		oic.r.windowcovering	ocf.liftvelocity = Velocity-Lift	N/A
Windowcoveringtype		oic.r.windowcovering	if WindowCoveringType=0x00, ocf.windowcoveringtype= Rollershade.if WindowCoveringType=0x01, ocf.windowcoveringtype= RollerShade-2 Motor.if WindowCoveringType=0x02, ocf.windowcoveringtype= RollerShade-Exterior.if WindowCoveringType=0x03, ocf.windowcoveringtype= RollerShade-Exterior-2 Motor.if WindowCoveringType=0x04, ocf.windowcoveringtype= Drapery.if WindowCoveringType=0x05, ocf.windowcoveringtype= Awning.if WindowCoveringType=0x06, ocf.windowcoveringtype= Shutter.if WindowCoveringType=0x07, ocf.windowcoveringtype= Tilt Blind - Tilt Only.if WindowCoveringType=0x08, ocf.windowcoveringtype= Tilt Blind & Lift and Tilt.if WindowCoveringType=0x09, ocf.windowcoveringtype= Projector Screen.	N/A
Config/Status		oic.r.windowcovering	if Config/Status =xxxxxx0, ocf.configstatus.operational = falseif Config/Status =xxxxxx1, ocf.configstatus.operational = trueif Config/Status =xxxxxx0x, ocf.configstatus.online = falseif Config/Status =xxxxxx1x, 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.configstatuscontrollift = 'closedloop'if Config/Status =xxx0xxxx, ocf.configstatus.controllift = 'openloop'if Config/Status =xxx1xxxx, ocf.configstatus.controllift =	N/A

		'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= Deceleration Time-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
Acceleration Time-Lift	oic.r.windowcovering	ocf.liftaccelerationtime= Acceleration Time-Lift	N/A

1856 Table 48 provides the details of the Properties that are part of "zcl.windowcovering_conf.info".

1857 **Table 48 – The Properties of "zcl.windowcovering_conf.info".**

Zigbee name	Property	Type	Required	Description
Velocity-Lift		integer	no	Velocity associated with Lifting the Window Covering (cm/sec).
Windowcoveringtype		string	yes	Type of Window Covering(i.e., [Rollershade, RollerShade-2 Motor, RollerShade-Exterior, RollerShade-Exterior-2 Motor, Drapery, Awning, Shutter, Tilt Blind - Tilt Only, Tilt Blind - Lift and Tilt, Projector Screen])
Config/Status		integer	yes	x is a variable. Config/Status in Zigbee maps to configstatus in OCF. Data type of Config/Status in Zigbee is 8 bitmap (xxxxxxx) : bit 0 = Operational, bit 1 =

		<p>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.</p>
--	--	--

Deceleration Time-Lift	integer	no	Ramp down times associated with stopping the velocity setting (0.1sec).
Mode	integer	yes	x is a variable. Mode in Zigbee maps to mode in OCF. Data type of Mode in Zigbee is 8 bitmap (xxxxxxx) : bit 0 = Reversed Motor Direction, bit 1 = Calibration Mode, bit 2 = Maintenance Mode, bit 3 = LED. Reversed Motor Direction : 0 = motor direction is normal, 1 = motor direction is reversed. Calibration Mode : 0 = run in normal mode, 1 = run in calibration mode. Maintenance Mode : 0 = motor is running normally, 1 = motor is running in maintenance mode. LED: 0 = LEDs are off, 1 = LEDs will display feedback.
Acceleration Time-Lift	integer	no	Ramp up times to reaching the velocity setting (0.1sec).

1858 **9.21.3 Derived model definition**

```

1859 {
1860   "id": "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_conf.info.json#",
1861   "$schema": "http://json-schema.org/draft-04/schema#",
1862   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1863   "title": "Window Covering Cluster - Configuration - Information",
1864   "definitions": {
1865     "zcl.windowcovering_conf.info": {
1866       "type": "object",
1867       "properties": {
1868         "windowcoveringtype": {
1869           "type": "string",
1870           "description": "Type of Window Covering(i.e., [Rollershade, RollerShade-2 Motor,
1871 RollerShade-Exterior, RollerShade-Exterior-2 Motor, Drapery, Awning, Shutter, Tilt Blind - Tilt
1872 Only, Tilt Blind â€ Lift and Tilt, Projector Screen])",
1873           "x-ocf-conversion": {
1874             "x-ocf-alias": "oic.r.windowcovering",
1875             "x-to-ocf": [
1876               "if WindowCoveringType=0x00, ocf.windowcoveringtype= Rollershade.",
1877               "if WindowCoveringType=0x01, ocf.windowcoveringtype= RollerShade-2 Motor.",
1878               "if WindowCoveringType=0x02, ocf.windowcoveringtype= RollerShade-Exterior.",
1879               "if WindowCoveringType=0x03, ocf.windowcoveringtype= RollerShade-Exterior-2 Motor.",
1880               "if WindowCoveringType=0x04, ocf.windowcoveringtype= Drapery.",
1881               "if WindowCoveringType=0x05, ocf.windowcoveringtype= Awning.",
1882               "if WindowCoveringType=0x06, ocf.windowcoveringtype= Shutter.",
1883               "if WindowCoveringType=0x07, ocf.windowcoveringtype= Tilt Blind - Tilt Only.",
1884               "if WindowCoveringType=0x08, ocf.windowcoveringtype= Tilt Blind â€ Lift and Tilt.",
1885               "if WindowCoveringType=0x09, ocf.windowcoveringtype= Projector Screen."
1886             ],
1887             "x-from-ocf": [
1888               "N/A"
1889             ]
1890           }
1891         }
1892       }
1893     }
1894   }

```

```

1891     },
1892     "Config/Status": {
1893         "type": "integer",
1894         "description": "x is a variable. Config/Status in Zigbee maps to configstatus in OCF.
1895 Data type of Config/Status in Zigbee is 8 bitmap (xxxxxxx) : bit 0 = Operational, bit 1 = Online,
1896 bit 2 = Reversal, bit 3 = Control-Lift, bit 4 = Control-Tilt, bit 5 = Encoder-Lift, bit 6 =
1897 Encoder-Tilt. Operational: This status bit defines if the Window Covering is operational. 0 = Not
1898 Operational, 1 = Operational. Online: This status bit defines if the Window Covering is enabled for
1899 transmitting over the ZigBee network. 0 = Not Online, 1 = Online. Reversal: This status bit
1900 identifies if the direction of rotation for the Window Covering has been reversed in order for
1901 Open/Up commands to match the physical installation condition. 0 = Commands are normal, 1 = Open/Up
1902 Commands have been reversed. Control Lift: This status bit identifies if the window covering
1903 supports Open Loop or Closed Loop Lift Control. 0 = Lift control is Open Loop, 1 = Lift control is
1904 Closed. Control Tilt: This status bit identifies if the window covering supports Open Loop or
1905 Closed Loop Tilt Control. 0 = Tilt control is Open Loop, 1 = Tilt control is Closed. Encoder Lift:
1906 This status bit identifies if a Closed Loop Controlled Window Covering is employing an encoder for
1907 positioning the height of the window covering. 0 = Timer Controlled, 1 = Encoder Controlled.
1908 Encoder Tilt: This status bit identifies if a Closed Loop Controlled Window Covering is employing
1909 an encoder for tilting the window covering. 0 = Timer Controlled, 1 = Encoder Controlled.",
1910     "x-ocf-conversion": {
1911         "x-ocf-alias": "oic.r.windowcovering",
1912         "x-to-ocf": [
1913             "if Config/Status =xxxxxxx0, ocf.configstatus.operational = false",
1914             "if Config/Status =xxxxxxx1, ocf.configstatus.operational = true",
1915             "if Config/Status =xxxxxxx0x, ocf.configstatus.online = false",
1916             "if Config/Status =xxxxxxx1x, ocf.configstatus.online = true",
1917             "if Config/Status =xxxxxx0xx, ocf.configstatus.rotationdirection = 'normal'",
1918             "if Config/Status =xxxxxx1xx, ocf.configstatus.rotationdirection = 'reversed'",
1919             "if Config/Status =xxxx0xxx, ocf.configstatus.controllift = 'openloop'",
1920             "if Config/Status =xxxx1xxx, ocf.configstatus.controllift = 'closedloop'",
1921             "if Config/Status =xxx0xxxx, ocf.configstatus.controltilt = 'openloop'",
1922             "if Config/Status =xxx1xxxx, ocf.configstatus.controltilt = 'closedloop'",
1923             "if Config/Status =xx0xxxxx, ocf.configstatus.closedloopliftcontrol = 'timer'",
1924             "if Config/Status =xx1xxxxx, ocf.configstatus.closedloopliftcontrol = 'encoder'",
1925             "if Config/Status =x0xxxxxx, ocf.configstatus.closedlooptiltcontrol = 'timer'",
1926             "if Config/Status =x1xxxxxx, ocf.configstatus.closedlooptiltcontrol = 'encoder'"
1927         ],
1928         "x-from-ocf": [
1929             "N/A"
1930         ]
1931     }
1932 },
1933 "Mode": {
1934     "type": "integer",
1935     "description": "x is a variable. Mode in Zigbee maps to mode in OCF. Data type of Mode in
1936 Zigbee is 8 bitmap (xxxxxxx) : bit 0 = Reversed Motor Direction, bit 1 = Calibration Mode, bit 2 =
1937 Maintenance Mode, bit 3 = LED. Reversed Motor Direction : 0 = motor direction is normal, 1 = motor
1938 direction is reversed. Calibration Mode : 0 = run in normal mode, 1 = run in calibration mode.
1939 Maintenance Mode : 0 = motor is running normally, 1 = motor is running in maintenance mode. LED: 0
1940 = LEDs are off, 1 = LEDs will display feedback.",
1941     "x-ocf-conversion": {
1942         "x-ocf-alias": "oic.r.windowcovering",
1943         "x-to-ocf": [
1944             "if Mode =xxxxxxx0, ocf.mode.motordirection = false",
1945             "if Mode =xxxxxxx1, ocf.mode.motordirection = true",
1946             "if Mode =xxxxxxx0x, ocf.mode.calibration = false",
1947             "if Mode =xxxxxxx1x, ocf.mode.calibration = true",
1948             "if Mode =xxxxxx0xx, ocf.mode.maintenance = false",
1949             "if Mode =xxxxxx1xx, ocf.mode.maintenance = true",
1950             "if Mode =xxxxx0xxx, ocf.mode.ledfeedback = false",
1951             "if Mode =xxxxx1xxx, ocf.mode.ledfeedback = true"
1952         ],
1953         "x-from-ocf": [
1954             "N/A"
1955         ]
1956     }
1957 },
1958 "Velocity-Lift": {
1959     "type": "integer",
1960     "description": "Velocity associated with Lifting the Window Covering (cm/sec).",
1961     "x-ocf-conversion": {

```



```

1962         "x-ocf-alias": "oic.r.windowcovering",
1963         "x-to-ocf": [
1964             "ocf.liftvelocity = Velocity-Lift"
1965         ],
1966         "x-from-ocf": [
1967             "N/A"
1968         ]
1969     }
1970 },
1971 "Acceleration Time-Lift": {
1972     "type": "integer",
1973     "description": "Ramp up times to reaching the velocity setting (0.1sec).",
1974     "x-ocf-conversion": {
1975         "x-ocf-alias": "oic.r.windowcovering",
1976         "x-to-ocf": [
1977             "ocf.liftaccelerationtime= Acceleration Time-Lift"
1978         ],
1979         "x-from-ocf": [
1980             "N/A"
1981         ]
1982     }
1983 },
1984 "Deceleration Time-Lift": {
1985     "type": "integer",
1986     "description": "Ramp down times associated with stoping the velocity setting (0.1sec).",
1987     "x-ocf-conversion": {
1988         "x-ocf-alias": "oic.r.windowcovering",
1989         "x-to-ocf": [
1990             "ocf.liftdecelerationtime= Deceleration Time-Lift"
1991         ],
1992         "x-from-ocf": [
1993             "N/A"
1994         ]
1995     }
1996 }
1997 }
1998 }
1999 },
2000 "type": "object",
2001 "allOf": [
2002     {"$ref": "#/definitions/zcl.windowcovering_conf.info"}
2003 ],
2004 "required": [ "Windowcoveringtype", "Config/Status", "Mode" ]
2005 }
2006

```

2007 9.22 Window Covering Cluster - Lift Percentage - Control

2008 9.22.1 Derived model

2009 The derived model: "zcl.windowcovering_liftpercentage.control.gotoliftpercentage".

2010 9.22.2 Property definition

2011 Table 49 provides the detailed per Property mapping for
2012 "zcl.windowcovering_liftpercentage.control.gotoliftpercentage".

2013 **Table 49 – The Property mapping for**
2014 **"zcl.windowcovering_liftpercentage.control.gotoliftpercentage".**

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

2015 Table 50 provides the details of the Properties that are part of
 2016 "zcl.windowcovering_liftpercentage.control.gotoliftpercentage".

2017 **Table 50 – The Properties of**
 2018 **"zcl.windowcovering_liftpercentage.control.gotoliftpercentage".**

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

2019 **9.22.3 Derived model definition**

```

2020 {
2021   "id":
2022   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_liftpercentage.control.json#"
2023   ,
2024   "$schema": "http://json-schema.org/draft-04/schema#",
2025   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2026   "title": "Window Covering Cluster - Lift Percentage - Control",
2027   "definitions": {
2028     "zcl.windowcovering_liftpercentage.control.gotoliftpercentage": {
2029       "properties": {
2030         "percentageliftvalue": {
2031           "type": "integer",
2032           "description": "Adjust the window at the percentage lift value.",
2033           "x-ocf-conversion": {
2034             "x-ocf-alias": "oic.r.openlevel",
2035             "x-from-ocf": [
2036               "if ocf.openLevel is updated, percentage lift value = ocf.openLevel.",
2037               "zcl.command.windowcovering::gotoliftpercentage(percentageliftvalue)"
2038             ],
2039             "x-to-ocf": [
2040               "N/A"
2041             ]
2042           }
2043         }
2044       }
2045     }
2046   },
2047   "type": "object",
2048   "allof": [
2049     {"$ref": "#/definitions/zcl.windowcovering_liftpercentage.control.gotoliftpercentage"}
2050   ]
2051 }
2052 
```

2053 **9.23 Window Covering Cluster - Lift Percentage - Information**

2054 **9.23.1 Derived model**

2055 The derived model: "zcl.windowcovering_liftpercentage.info".

2056 **9.23.2 Property definition**

2057 Table 51 provides the detailed per Property mapping for "zcl.windowcovering_liftpercentage.info".

2058 **Table 51 – 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

2059 Table 52 provides the details of the Properties that are part of
 2060 "zcl.windowcovering_liftpercentage.info".

Table 52 – 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

2062 9.23.3 Derived model definition

```

2063 {
2064   "id":
2065   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_liftpercentage.info.json#",
2066   "$schema": "http://json-schema.org/draft-04/schema#",
2067   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2068   "title": "Window Covering Cluster - Lift Percentage - Information",
2069   "definitions": {
2070     "zcl.windowcovering_liftpercentage.info": {
2071       "type": "object",
2072       "properties": {
2073         "CurrentPositionLiftPercentage": {
2074           "type": "integer",
2075           "description": "Position as a percentage between InstalledOpenLimit-Lift and
2076 InstalledClosedLimit-Lift",
2077           "x-ocf-conversion": {
2078             "x-ocf-alias": "oic.r.openlevel",
2079             "x-to-ocf": [
2080               "ocf.openLevel= CurrentPositionLiftPercentage"
2081             ],
2082             "x-from-ocf": [
2083               "N/A"
2084             ]
2085           }
2086         }
2087       }
2088     },
2089   },
2090   "type": "object",
2091   "allOf": [
2092     {"$ref": "#/definitions/zcl.windowcovering_liftpercentage.info"}
2093   ],
2094   "required": ["CurrentPositionLiftPercentage"]
2095 }
2096

```

2097 9.24 Window Covering Cluster - Lift Position - Control

2098 9.24.1 Derived model

2099 The derived model: "zcl.windowcovering_liftposition.control.gotoliftvalue".

2100 9.24.2 Property definition

2101 Table 53 provides the detailed per Property mapping for
 2102 "zcl.windowcovering_liftposition.control.gotoliftvalue".

2103 **Table 53 – The Property mapping for**
 2104 **"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)

2105 Table 54 provides the details of the Properties that are part of
 2106 "zcl.windowcovering_liftposition.control.gotoliftvalue".

2107 **Table 54 – The Properties of "zcl.windowcovering_liftposition.control.gotoliftvalue".**

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

2108 **9.24.3 Derived model definition**

```

2109 {
2110   "id":
2111   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_liftposition.control.json#",
2112   "$schema": "http://json-schema.org/draft-04/schema#",
2113   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2114   "title": "Window Covering Cluster - Lift Position - Control",
2115   "definitions": {
2116     "zcl.windowcovering_liftposition.control.gotoliftvalue": {
2117       "properties": {
2118         "liftvalue": {
2119           "type": "integer",
2120           "description": "Adjust the window at the lift value.",
2121           "x-ocf-conversion": {
2122             "x-ocf-alias": "oic.r.openlevel",
2123             "x-from-ocf": [
2124               "if ocf.openLevel is updated, lift value= ocf.openLevel.",
2125               "zcl.command.windowcovering::gotoliftvalue(liftvalue)"
2126             ],
2127             "x-to-ocf": [
2128               "N/A"
2129             ]
2130           }
2131         }
2132       }
2133     }
2134   },
2135   "type": "object",
2136   "allof": [
2137     {"$ref": "#/definitions/zcl.windowcovering_liftposition.control.gotoliftvalue"}
2138   ]
2139 }
2140

```

2141 **9.25 Window Covering Cluster - Lift Position - Information**

2142 **9.25.1 Derived model**

2143 The derived model: "zcl.windowcovering_liftposition.info".

2144 **9.25.2 Property definition**

2145 Table 55 provides the detailed per Property mapping for "zcl.windowcovering_liftposition.info".

2146 **Table 55 – The Property mapping for "zcl.windowcovering_liftposition.info".**

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

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

2147 Table 56 provides the details of the Properties that are part of
2148 "zcl.windowcovering_liftposition.info".

2149 **Table 56 – The Properties of "zcl.windowcovering_liftposition.info".**

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

2150 9.25.3 Derived model definition

```

2151 {
2152   "id":
2153   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_liftposition.info.json#",
2154   "$schema": "http://json-schema.org/draft-04/schema#",
2155   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2156   "title": "Window Covering Cluster - Lift Position - Information",
2157   "definitions": {
2158     "zcl.windowcovering_liftposition.info": {
2159       "type": "object",
2160       "properties": {
2161         "InstalledClosedLimit-Lift": {
2162           "type": "integer",
2163           "description": "Close limit for lifting the Window Covering (cm)",
2164           "x-ocf-conversion": {
2165             "x-ocf-alias": "oic.r.openlevel",
2166             "x-to-ocf": [
2167               "ocf.range[0]= InstalledClosedLimit-Lift"
2168             ],
2169             "x-from-ocf": [
2170               "N/A"
2171             ]
2172           }
2173         },
2174         "InstalledOpenLimit-Lift": {
2175           "type": "integer",
2176           "description": "Open limit for lifting the Window Covering (cm)",
2177           "x-ocf-conversion": {
2178             "x-ocf-alias": "oic.r.openlevel",
2179             "x-to-ocf": [
2180               "ocf.range[1]= InstalledOpenLimit-Lift"
2181             ],
2182             "x-from-ocf": [
2183               "N/A"
2184             ]
2185           }
2186         },
2187         "CurrentPosition-Lift": {
2188           "type": "integer",
2189           "description": "Position of Window Covering from the top of the shade (cm)",
2190           "x-ocf-conversion": {
2191             "x-ocf-alias": "oic.r.openlevel",
2192             "x-to-ocf": [
2193               "ocf.openLevel= CurrentPosition-Lift"
2194             ],

```

```

2195         "x-from-ocf": [
2196             "N/A"
2197         ]
2198     }
2199 }
2200 }
2201 }
2202 },
2203 "type": "object",
2204 "allOf": [
2205     {"$ref": "#/definitions/zcl.windowcovering_liftposition.info"}
2206 ],
2207 "required": [ "InstalledClosedLimit-Lift", "InstalledOpenLimit-Lift", "CurrentPosition-Lift" ]
2208 }
2209 }

```

2210 9.26 Window Covering Cluster - Tilt Percentage - Control

2211 9.26.1 Derived model

2212 The derived model: "zcl.windowcovering_tiltpercentage.control.gototiltpercentage".

2213 9.26.2 Property definition

2214 Table 57 provides the detailed per Property mapping for
 2215 "zcl.windowcovering_tiltpercentage.control.gototiltpercentage".

2216 **Table 57 – The Property mapping for**
 2217 **"zcl.windowcovering_tiltpercentage.control.gototiltpercentage".**

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

2218 Table 58 provides the details of the Properties that are part of
 2219 "zcl.windowcovering_tiltpercentage.control.gototiltpercentage".

2220 **Table 58 – The Properties of**
 2221 **"zcl.windowcovering_tiltpercentage.control.gototiltpercentage".**

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

2222 9.26.3 Derived model definition

```

2223 {
2224     "id":
2225     "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_tiltpercentage.control.json#"
2226     ,
2227     "$schema": "http://json-schema.org/draft-04/schema#",
2228     "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2229     "title": "Window Covering Cluster - Tilt Percentage - Control",
2230     "definitions": {
2231         "zcl.windowcovering_tiltpercentage.control.gototiltpercentage": {
2232             "properties": {
2233                 "percentagetiltvalue": {
2234                     "type": "integer",
2235                     "description": "Adjust the window at the percentage tilt value.",
2236                     "x-ocf-conversion": {
2237                         "x-ocf-alias": "oic.r.openlevel",
2238                         "x-from-ocf": [

```

```

2239         "if ocf.openLevel is updated, percentage tilt value = ocf.openLevel.",
2240         "zcl.command.windowcovering::gototiltpercentage(percentage tiltvalue)"
2241     ],
2242     "x-to-ocf": [
2243         "N/A"
2244     ]
2245 }
2246 }
2247 }
2248 }
2249 },
2250 "type": "object",
2251 "allOf": [
2252     {"$ref": "#/definitions/zcl.windowcovering_tiltpercentage.control.gototiltpercentage"}
2253 ]
2254 }
2255 }

```

2256 9.27 Window Covering Cluster - Tilt Percentage - Information

2257 9.27.1 Derived model

2258 The derived model: "zcl.windowcovering_tiltpercentage.info".

2259 9.27.2 Property definition

2260 Table 59 provides the detailed per Property mapping for "zcl.windowcovering_tiltpercentage.info".

2261 **Table 59 – 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

2262 Table 60 provides the details of the Properties that are part of
2263 "zcl.windowcovering_tiltpercentage.info".

2264 **Table 60 – The Properties of "zcl.windowcovering_tiltpercentage.info".**

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

2265 9.27.3 Derived model definition

```

2266 {
2267     "id":
2268     "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_tiltpercentage.info.json#",
2269     "$schema": "http://json-schema.org/draft-04/schema#",
2270     "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2271     "title": "Window Covering Cluster - Tilt Percentage - Information",
2272     "definitions": {
2273         "zcl.windowcovering_tiltpercentage.info": {
2274             "type": "object",
2275             "properties": {
2276                 "CurrentPositionTiltPercentage": {
2277                     "type": "integer",
2278                     "description": "Tilt position as a percentage",
2279                     "x-ocf-conversion": {
2280                         "x-ocf-alias": "oic.r.openlevel",
2281                         "x-to-ocf": [
2282                             "ocf.openlevel=CurrentPositionTiltPercentage"
2283                         ],
2284                         "x-from-ocf": [
2285                             "N/A"
2286                         ]

```

```

2287     }
2288   }
2289 }
2290 }
2291 },
2292 "type": "object",
2293 "allOf": [
2294   {"$ref": "#/definitions/zcl.windowcovering_tiltpercentage.info"}
2295 ],
2296 "required": ["CurrentPositionTiltPercentage"]
2297 }
2298

```

2299 9.28 Window Covering Cluster - Tilt Position - Control

2300 9.28.1 Derived model

2301 The derived model: "zcl.windowcovering_tiltposition.control.gototiltvalue".

2302 9.28.2 Property definition

2303 Table 61 provides the detailed per Property mapping for
 2304 "zcl.windowcovering_tiltposition.control.gototiltvalue".

2305 **Table 61 – The Property mapping for**
 2306 **"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)

2307 Table 62 provides the details of the Properties that are part of
 2308 "zcl.windowcovering_tiltposition.control.gototiltvalue".

2309 **Table 62 – The Properties of "zcl.windowcovering_tiltposition.control.gototiltvalue".**

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

2310 9.28.3 Derived model definition

```

2311 {
2312   "id":
2313   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_tiltposition.control.json#",
2314   "$schema": "http://json-schema.org/draft-04/schema#",
2315   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2316   "title": "Window Covering Cluster - Tilt Position - Control",
2317   "definitions": {
2318     "zcl.windowcovering_tiltposition.control.gototiltvalue": {
2319       "properties": {
2320         "tiltvalue": {
2321           "type": "integer",
2322           "description": "Adjust the window at the tilt value.",
2323           "x-ocf-conversion": {
2324             "x-ocf-alias": "oic.r.openlevel",
2325             "x-from-ocf": [
2326               "if ocf.openLevel is updated, tiltvalue= ocf.openLevel.",
2327               "zb.command.windowcovering::gototiltvalue(tiltvalue)"
2328             ],
2329             "x-to-ocf": [
2330               "N/A"
2331             ]
2332           }
2333         }
2334       }
2335     }
2336   }

```



```

2332     }
2333   }
2334 }
2335 }
2336 },
2337 "type": "object",
2338 "allOf": [
2339   {"$ref": "#/definitions/zcl.windowcovering_tiltposition.control.gototiltvalue"}
2340 ]
2341 }
2342

```

2343 **9.29 Window Covering Cluster - Tilt Position - Information**

2344 **9.29.1 Derived model**

2345 The derived model: "zcl.windowcovering_tiltposition.info".

2346 **9.29.2 Property definition**

2347 Table 63 provides the detailed per Property mapping for "zcl.windowcovering_tiltposition.info".

2348 **Table 63 – The Property mapping for "zcl.windowcovering_tiltposition.info".**

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

2349 Table 64 provides the details of the Properties that are part of
 2350 "zcl.windowcovering_tiltposition.info".

2351 **Table 64 – The Properties of "zcl.windowcovering_tiltposition.info".**

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

2352 **9.29.3 Derived model definition**

```

2353 {
2354   "id":
2355   "http://openinterconnect.org/zigbeemapping/schemas/zcl.windowcovering_tiltposition.info.json#",
2356   "$schema": "http://json-schema.org/draft-04/schema#",
2357   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
2358   "title": "Window Covering Cluster - Tilt Position - Information",
2359   "definitions": {
2360     "zcl.windowcovering_tiltposition.info": {
2361       "type": "object",
2362       "properties": {
2363         "InstalledOpenLimit-Tilt": {
2364           "type": "integer",
2365           "description": "Close limit for tilting the Window Covering (0.1 degree)",
2366           "x-ocf-conversion": {
2367             "x-ocf-alias": "oic.r.openlevel",
2368             "x-to-ocf": [
2369               "ocf.range[0] = InstalledClosedLimit-Tilt"

```

```

2370         ],
2371         "x-from-ocf": [
2372             "N/A"
2373         ]
2374     },
2375 },
2376 "InstalledOpenLimit-Tilt": {
2377     "type": "integer",
2378     "description": "Open limit for tilting the Window Covering (0.1 degree)",
2379     "x-ocf-conversion": {
2380         "x-ocf-alias": "oic.r.openlevel",
2381         "x-to-ocf": [
2382             "ocf.range[1]= InstalledOpenLimit-Tilt"
2383         ],
2384         "x-from-ocf": [
2385             "N/A"
2386         ]
2387     },
2388 },
2389 "CurrentPosition-Tilt ": {
2390     "type": "integer",
2391     "description": "Tilt position of Window Covering from open (0.1 degree)",
2392     "x-ocf-conversion": {
2393         "x-ocf-alias": "oic.r.openlevel",
2394         "x-to-ocf": [
2395             "ocf.openlevel= CurrentPosition-Tilt"
2396         ],
2397         "x-from-ocf": [
2398             "N/A"
2399         ]
2400     },
2401 },
2402 },
2403 },
2404 },
2405 "type": "object",
2406 "allOf": [
2407     {"$ref": "#/definitions/zcl.windowcovering_tiltposition.info"}
2408 ],
2409 "required": [ "InstalledClosedLimit-Tilt", "InstalledOpenLimit-Tilt", "CurrentPosition-Tilt" ]
2410 }
2411
2412

```