

OCF Resource to Z-Wave Mapping Specification

VERSION 2.2.1 | December 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 abbreviated terms	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.....	3
34	5.2.3	Value Assignment	3
35	5.2.4	Property Naming	3
36	5.2.5	Range	3
37	5.2.6	Arrays	3
38	5.2.7	Default Mapping	3
39	5.2.8	Conditional Mapping.....	3
40	5.2.9	Method Invocation.....	4
41	6	Z-Wave Translation	4
42	6.1	Operational scenarios.....	4
43	6.1.1	Introduction	4
44	6.1.2	Overview of OCF-Z-Wave bridging	4
45	6.1.3	Use case for OCF Client and Z-Wave server	4
46	6.2	Requirements specific to Z-Wave Bridging Function	5
47	6.2.1	Requirements specific to Z-Wave	5
48	6.2.2	Exposing Z-Wave servers to OCF clients	5
49	7	Device Type Mapping.....	12
50	7.1	Introduction	12
51	7.2	Z-Wave Device Types to OCF Device Types	12
52	8	Resource to Command Class Mapping	13
53	8.1	Introduction	13
54	8.2	Z-Wave Command Classes to OCF Resources	13
55	8.2.1	Introduction	13
56	8.2.2	Battery Command Class Mapping.....	14
57	8.2.3	Binary Switch Command Class Mapping.....	14
58	8.2.4	Door Lock Command Class Mapping.....	14
59	8.2.5	Multilevel Sensor Command Class Mapping	14
60	8.2.6	Multilevel Switch Command Class Mapping.....	15
61	8.2.7	Notification Command Class Mapping	15
62	8.2.8	User Code Command Class Mapping	15
63	9	Detailed Mapping APIs.....	16

64	9.1	Battery Command Class	16
65	9.1.1	Derived model	16
66	9.1.2	Property definition	16
67	9.1.3	Derived model definition	17
68	9.2	Binary Switch Command Classes	17
69	9.2.1	Derived model	17
70	9.2.2	Property definition	17
71	9.2.3	Derived model definition	18
72	9.3	Door Lock Command Class	18
73	9.3.1	Derived model	18
74	9.3.2	Property definition	18
75	9.3.3	Derived model definition	19
76	9.4	Multilevel Sensor Command Class Carbon Dioxide	19
77	9.4.1	Derived model	19
78	9.4.2	Property definition	19
79	9.4.3	Derived model definition	20
80	9.5	Multilevel Sensor Command Class Carbon Monoxide	21
81	9.5.1	Derived model	21
82	9.5.2	Property definition	21
83	9.5.3	Derived model definition	22
84	9.6	Multilevel Sensor Command Class Smoke Density	24
85	9.6.1	Derived model	24
86	9.6.2	Property definition	24
87	9.6.3	Derived model definition	24
88	9.7	Multilevel Sensor Command Class Water Flow	26
89	9.7.1	Derived model	26
90	9.7.2	Property definition	26
91	9.7.3	Derived model definition	26
92	9.8	Multilevel Switch Command Class	28
93	9.8.1	Derived model	28
94	9.8.2	Property definition	28
95	9.8.3	Derived model definition	28
96	9.9	Notification Command Class	29
97	9.9.1	Derived model	29
98	9.9.2	Property definition	29
99	9.9.3	Derived model definition	30
100	9.10	User Code Command Class	32
101	9.10.1	Derived model	32
102	9.10.2	Property definition	33
103	9.10.3	Derived model definition	33
104			
105			

106
107
108
109

Figures

Figure 1 – OCF Z-Wave Bridge Platform and Components	4
Figure 2 – OCF Client and Z-Wave Server.....	5

Tables

111	Table 3 – Z-Wave Device & Command Class – OCF Device & Resource mapping.....	7
112	Table 4 – "oic.wk.d" Resource Type definition	8
113	Table 5 – "oic.wk.con" Resource Type definition.....	10
114	Table 6 – "oic.wk.p" Resource Type definition	11
115	Table 7 – Z-Wave to OCF Device Type Mapping	12
116	Table 8 – Z-Wave Command Class to OCF Resource Type Mapping.....	13
117	Table 9 – Command Class to Resource Summary	16
118	Table 10 – The Property mapping for "zwave.operation.batterycommandclass".	16
119	Table 11 – The Properties of "zwave.operation.batterycommandclass".....	17
120	Table 12 – The Property mapping for "zwave.operation.binaryswitchcommandclass".....	17
121	Table 13 – The Properties of "zwave.operation.binaryswitchcommandclass".	18
122	Table 14 – The Property mapping for "zwave.operation.doorlockcommandclass".....	18
123	Table 15 – The Properties of "zwave.operation.doorlockcommandclass".	19
124	Table 16 – The Property mapping for	
125	"zwave.operation.multilevelsensorcommandclasscarbondioxide".....	19
126	Table 17 – The Properties of	
127	"zwave.operation.multilevelsensorcommandclasscarbondioxide".....	20
128	Table 18 – The Property mapping for	
129	"zwave.operation.multilevelsensorcommandclasscarbonmonoxide".	22
130	Table 19 – The Properties of	
131	"zwave.operation.multilevelsensorcommandclasscarbonmonoxide".	22
132	Table 20 – The Property mapping for	
133	"zwave.operation.multilevelsensorcommandclasssmokedensity".....	24
134	Table 21 – The Properties of	
135	"zwave.operation.multilevelsensorcommandclasssmokedensity".....	24
136	Table 22 – The Property mapping for	
137	"zwave.operation.multilevelsensorcommandclasswaterflow".	26
138	Table 23 – The Properties of "zwave.operation.multilevelsensorcommandclasswaterflow"....	26
139	Table 24 – The Property mapping for "zwave.operation.multilevelswitchcommandclass".	28
140	Table 25 – The Properties of "zwave.operation.multilevelswitchcommandclass".	28
141	Table 26 – The Property mapping for "zwave.operation.notificationcommandclass".	29
142	Table 27 – The Properties of "zwave.operation.notificationcommandclass".....	30
143	Table 28 – The Property mapping for "zwave.operation.usercodecommandclass".....	33
144	Table 29 – The Properties of "zwave.operation.usercodecommandclass".	33
145		

146 **1 Scope**

147 This document provides detailed mapping information between Z-Wave and OCF defined
148 Resources.

149 **2 Normative references**

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

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

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

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

166 Derived Models for Interoperability between IoT Ecosystems, Stevens & Merriam, March 2016
167 [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)
168 [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)

169 Z-Wave Plus Device and Command Class Types Specification
170 [https://www.silabs.com/documents/login/miscellaneous/SDS11847-Z-Wave-Plus-Device-Type-](https://www.silabs.com/documents/login/miscellaneous/SDS11847-Z-Wave-Plus-Device-Type-Specification.pdf)
171 [Specification.pdf](https://www.silabs.com/documents/login/miscellaneous/SDS11847-Z-Wave-Plus-Device-Type-Specification.pdf)

172
173 Z-Wave Plus v2 Device Type Specification
174 [https://www.silabs.com/documents/login/miscellaneous/SDS14224-Z-Wave-Plus-v2-Device-Type-](https://www.silabs.com/documents/login/miscellaneous/SDS14224-Z-Wave-Plus-v2-Device-Type-Specification.pdf)
175 [Specification.pdf](https://www.silabs.com/documents/login/miscellaneous/SDS14224-Z-Wave-Plus-v2-Device-Type-Specification.pdf)

176

177 **3 Terms, definitions, symbols and abbreviated terms**

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

180 ISO and IEC maintain terminological databases for use in standardization at the following
181 addresses:

- 182 – ISO Online browsing platform: available at <https://www.iso.org/obp>
- 183 – IEC Electropedia: available at <http://www.electropedia.org/>

184 **3.1 Terms and definitions**

185 **3.1.1**

186 **Command Class**

187 a collection of commands used for controlling, querying, and reporting information corresponding
188 to specific function supported by a Z-Wave device.

189 **4 Document conventions and organization**

190 **4.1 Conventions**

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

195 **4.2 Notation**

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

198 Required (or shall or mandatory).

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

202 Recommended (or should).

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

210 Allowed (or allowed).

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

214 Conditionally allowed (CA)

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

217 Conditionally required (CR)

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

221 DEPRECATED

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

228 Strings that are to be taken literally are enclosed in "double quotes".

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

230 **5 Theory of Operation**

231 **5.1 Interworking Approach**

232 The interworking between Z-Wave defined Command Classes and OCF defined Resources is
233 modelled using the derived model syntax described in Derived Models for Interoperability between
234 IoT Ecosystems.

235 **5.2 Mapping Syntax**

236 **5.2.1 Introduction**

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

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

244 **5.2.2 General**

245 All statements are terminated with a carriage return.

246 **5.2.3 Value Assignment**

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

249 **5.2.4 Property Naming**

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

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

255 **5.2.5 Range**

256 The range on the OCF side is fixed.

257 **5.2.6 Arrays**

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

260 **5.2.7 Default Mapping**

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

264 **5.2.8 Conditional Mapping**

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

266 If "condition", then "mapping".

267 is applied.

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

269 **5.2.9 Method Invocation**

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

274 **6 Z-Wave Translation**

275 **6.1 Operational scenarios**

276 **6.1.1 Introduction**

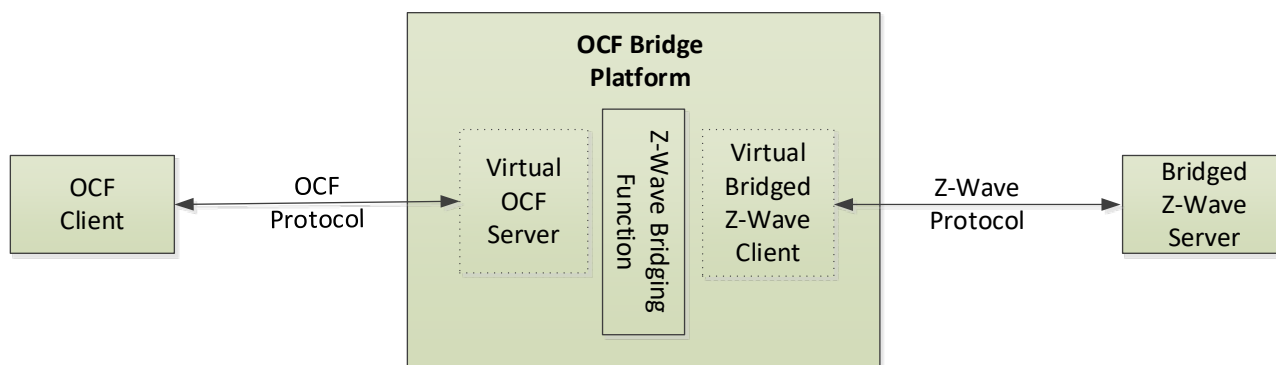
277 The overall goals are to:

- 278 – make Bridged Z-Wave Servers appear to OCF Clients as if they were native OCF Servers in
279 the local network or cloud environment

280 “Deep translation” between a specific Z-Wave device and an OCF Device is specified in clause 9.
281 “On-the-fly” translation is out of scope (refer to clause 5.1 “Deep translation” vs. “on-the-fly” of
282 ISO/IEC 30118-3:2019).

283 **6.1.2 Overview of OCF-Z-Wave bridging**

284 An OCF Z-Wave Bridge Platform provides the bridging function between an OCF Client and a
285 Bridged Z-Wave Server. The asymmetric bridging is applied to Z-Wave Bridging Function. Z-Wave
286 Bridging Function is performing the translation to or from the Z-Wave Protocol. The Z-Wave Bridge
287 Platform exposes Bridged Z-Wave Servers to OCF Clients and any OCF Cloud. A Bridged Z-Wave
288 Server provides Z-Wave specific data via the Z-Wave protocol for a Virtual Bridged Z-Wave Client.
289 Figure 1 presents the overview of an OCF Z-Wave Bridge Platform and its general topology.



290

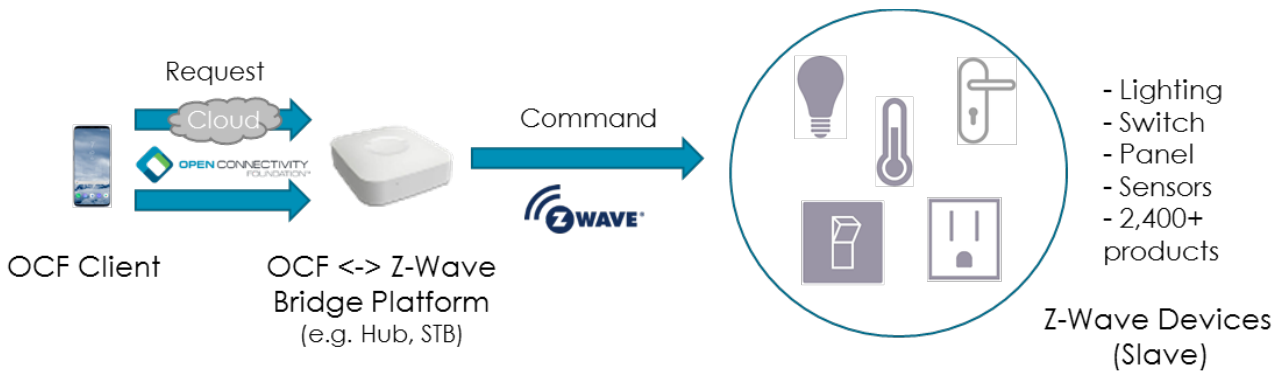
291

Figure 1 – OCF Z-Wave Bridge Platform and Components

292 **6.1.3 Use case for OCF Client and Z-Wave server**

293 A use case for an OCF Client and Z-Wave Server is presented in Figure 2. A smartphone device
294 acting as the OCF Client is allowed to send commands for controlling, querying and reporting the
295 information of Z-Wave devices via an OCF Z-Wave Bridge Platform. For that, Z-Wave Server
296 devices such as door locks with a keypad and light dimmer switch are represented as virtual OCF
297 Z-Wave server devices on an OCF Z-Wave Bridge Platform. Any connectivity that OCF supports

298 is used to communicate between OCF Client and an OCF Z-Wave Bridge. Furthermore, an OCF
 299 Client can also communicate with an OCF Z-Wave Bridge Platform via an OCF Cloud.



300

301 **Figure 2 – OCF Client and Z-Wave Server**

302 **6.2 Requirements specific to Z-Wave Bridging Function**

303 **6.2.1 Requirements specific to Z-Wave**

304 The version of Z-Wave device type for OCF Z-Wave Bridging shall be Z-Wave Plus or Z-Wave Plus
 305 v2. The Z-Wave Bridging Function shall act as Z-Wave Controller which sets up and performs
 306 maintenance operations such as inclusion and exclusion of devices in a Z-Wave network.

307 **6.2.2 Exposing Z-Wave servers to OCF clients**

308 **6.2.2.1 General**

309 The translation rule between Z-Wave and OCF data model is described in Table 1. The nature of
 310 how Z-Wave devices are structured may be different than how an OCF Device is structured. For
 311 example, Light Dimmer Switch is mapped to OCF Light with the device type "oic.d.light" and a
 312 Sensor – Multilevel and a Sensor – Notification is mapped to OCF Sensors with the Device Type
 313 "oic.d.sensor". A Z-Wave Command Class may be mapped to one or more OCF Resources. For
 314 instance, Multilevel Switch Command Class is mapped to OCF binary switch and dimming light.
 315 Each Command Class parameter is conditionally required to be mapped to a Property of an OCF
 316 Resource.

317 **Table 1 – Translation Rule between Z-Wave and OCF data model**

From Z-Wave	Mapping count	To OCF	Mapping count
Z- Wave Plus Device Type	n	OCF Device	1
Command Class	1	OCF Resource	n
Parameter	1	OCF Resource property	1

318

319 Table 2 is a mapping example of this rule.

Table 2 – Z-Wave → OCF mapping example (Light Dimmer Switch)

Z-Wave		OCF	
Z-Wave Plus Device Type	Light Dimmer Switch	OCF Device	"oic.d.light" (Light)
Command Class	Multilevel Switch Command Class (Multilevel Switch Set/Get/Report)	OCF Resource(s)	"oic.r.switch.binary" (Value)
			"oic.r.light.dimming" (dimmingSetting)
	Manufacturer Specific Command Class (Manufacturer Specific Get/Report)		"oic.wk.d" (Device) "oic.wk.p" (Platform)
	Version Command Class (Version Get/Report)		
	Z-Wave Plus Info Command Class (Z-Wave Plus Info Get/Report)		
Z-Wave Command Parameter	Value (255 or 0)	OCF Resource Property	"value" (True or False)
	Value (1-99)		"dimmingSetting" (Integer)

321 If Z-Wave Plus device, Z-Wave Command Class, Z-Wave Command Parameter are enlisted in the
 322 well-defined set as specified in OCF Z-Wave Data Model Mapping, Bridging Function shall follow
 323 the requirements for translating it to an OCF device, OCF resource or OCF resource property (i.e.,
 324 “deep translation”).

325 A Z-Wave Server device maps to a single OCF Device Type. The OCF Device Type is provided by
 326 using the Device identifier of the Z-Wave Server device. Z-Wave Bridging Function has a table
 327 which includes the mapping information between the Z-Wave device identifier and the OCF Device
 328 Type. Based on the table, the Z-Wave Bridging Function finds the Device Type according to the Z-
 329 Wave device identifier.

330 A Z-Wave device includes one or more Z-Wave Command Class. If a Z-Wave Command Class
 331 maps to resource type on a single OCF resource, there should be a single Virtual OCF Resource.
 332 If a Z-Wave Command Class maps to multiple OCF resource, an OCF resource may exist with an
 333 OCF Resource Type of ["oic.wk.col"] which is a Collection of links. The links in the collection are
 334 the Resources with translated Resource Types. The resource mapping between Z-Wave Server
 335 and OCF Resources is defined clause 8. The Z-Wave Bridging Function have a table which
 336 includes the mapping information between the identifier of Command Class and OCF Resource
 337 Type(s). After a virtual Bridged Z-Wave Client and Bridged Z-Wave Server device have done the
 338 inclusion procedure as specified in the Z-Wave Plus Device and Command Class Types
 339 Specification, a Z-Wave Bridging Function obtains the list of Command Class identifiers. Based
 340 upon the table, a Z-Wave Bridging Function finds the matched OCF Resource Type(s) according
 341 to the identifier of Z-Wave Command Class.

342 Since the Bridging Function knows all relationships between OCF Resources and Z-Wave servers,
 343 the path component of URI can be freely chosen. To maintain the relationship information and URI
 344 definition is implementation specific.

345 If a Z-Wave operation fails, the Bridging Function sends an appropriate OCF error response to the
 346 OCF Client. It constructs an appropriate OCF error message (e.g., diagnostic payload if using
 347 CoAP) from the Z-Wave enumerated status value and Z-Wave error message (if any), using the
 348 form "<error name>: <error message>", with the <error name> and <error message> taken from
 349 the Z-Wave error message and the error code for the OCF network set to an appropriate value.

350 **6.2.2.2 Translation for well-defined set**

351 Table 3 is the list of Z-Wave Plus device types which have corresponding OCF Resources.

352 **Table 3 – Z-Wave Device & Command Class – OCF Device & Resource mapping**

Z- Wave Plus Device	Z-Wave Command Class	OCF Resource Type	OCF Device Type	OCF Device Name
Light Dimmer Switch	Multilevel Switch Command Class	oic.r.switch.binary	oic.d.light	Light
	Multilevel Switch Command Class	oic.r.light.dimming		
	Manufacturer Specific Command Class Version Command Class Z-Wave Plus Info Command Class	oic.wk.d		
		oic.wk.p		
Door Lock – Keypad	Door Lock Command Class	oic.r.lock.status	oic.d.smartlock	Smart Lock
	User Code Command Class	oic.r.lock.code		
	Battery Command Class	oic.r.energy.battery.		
	Manufacturer Specific Command Class Version Command Class Z-Wave Plus Info Command Class	oic.wk.d		
		oic.wk.p		
On/Off Power Switch	Binary Switch Command Class	oic.r.switch.binary	oic.d.switch	Switch
	Battery Command Class	oic.r.energy.battery.		
	Manufacturer Specific Command Class Version Command Class Z-Wave Plus Info Command Class	oic.wk.d		
		oic.wk.p		
Sensor - Multilevel	Multilevel Sensor Command Class	oic.r.sensor.carbondioxide	oic.d.sensor	Generic Sensor
	Multilevel Sensor Command Class	oic.r.sensor.carbonmonoxide		
	Multilevel Sensor Command Class	oic.r.sensor.water		
	Multilevel Sensor Command Class	oic.r.sensor.smoke		
	Battery Command Class	oic.r.energy.battery.		
	Manufacturer Specific Command Class Version Command Class	oic.wk.d		
oic.wk.p				

	Z-Wave Plus Info Command Class			
Sensor - Notification	Notification Command Class	oic.r.sensor.carbondioxide	oic.d.sensor	Generic Sensor
	Notification Command Class	oic.r.sensor.carbonmonoxide		
	Notification Command Class	oic.r.sensor.water		
	Notification Command Class	oic.r.sensor.smoke		
	Battery Command Class	oic.r.energy.battery.		
	Manufacturer Specific Command Class Version Command Class Z-Wave Plus Info Command Class	oic.wk.d oic.wk.p		

353 Z-Wave Plus v2 device types which are equivalently mapped to the Z-Wave Plus device types that
354 supports deep translation are should be translated as specified in the table as well.

355 6.2.2.3 Exposing a Z-Wave Server as a Virtual OCF Server

356 Table 4 shows how OCF Device properties, as specified in ISO/IEC 30118-1:2018, shall be derived,
357 typically from fields of Command Parameter of Z-Wave Command Classes specified in Z-Wave
358 Plus Device and Command Class Types Specification. As specified in ISO/IEC 30118-2:2019, the
359 value of the "di" property of OCF Devices (including Virtual OCF Devices) shall be established as
360 part of on-boarding of that Virtual OCF Device.

361 **Table 4 – "oic.wk.d" Resource Type definition**

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory	From Z-Wave Field name	Z-Wave Description	Z-Wave Mandatory*
(Device) Name	n	Human friendly name For example, "Bob's Thermostat"	Y	Translate Product ID to Human friendly name based upon the Product ID/product name table within Z-Wave Controller	Product ID: a unique ID identifying the actual product as defined by the manufacturer for each product of a given product type. Defined in Manufacturer Specific Command Class	Product ID: Y
Spec Version	icv	Spec version of ISO/IEC 30118-1:2018 this device is implemented to, The syntax is "core.major.minor"]	Y	(none)	Bridge Platform should return its own value	
Device ID	di	Unique identifier for Device. This value shall be as defined in ISO/IEC 30118-2:2019for Device ID.	Y	(none)	Use as defined in ISO/IEC 30118-2:2019	

Protocol-Independent ID	piid	Unique identifier for OCF Device (UUID)	Y	(none)	Bridging Function should return a random-generated UUID as specified in the section 4.4 of IETF RFC 4122	
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)	Bridge Platform 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)		
Software Version	sv	Version of the device software.	N	Firmware 0 Version	Dedicated to the Z-Wave chip firmware as defined by the manufacturer which assigns a version number Defined in Version Command Class	N
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	Translate Manufacturer ID to Human friendly name based upon the Manufacturer ID/Manufacturer name table within Z-Wave Controller	Manufacturer ID: the unique ID identifying the manufacturer of the device. Defined in Manufacturer Specific Command Class	Y
Model Number	dmno	Model number as designated by manufacturer.	N	Product ID	A unique ID identifying the actual product as defined by the manufacturer for each product of a given product type. Defined in Manufacturer Specific Command Class	Y

362 Table 5 shows how OCF Device Configuration properties, as specified in ISO/IEC 30118-1:2018,
 363 shall be derived:

364 **Table 5 – "oic.wk.con" Resource Type definition**

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory	From Z-Wave Field name	Z-Wave Description	Z-Wave Mandatory*
(Device) Name	n	Human friendly name For example, "Bob's Thermostat"	Y	Translate Product ID to Human friendly name based upon the Product ID/product name table within Z-Wave Controller	Product ID: a unique ID identifying the actual product as defined by the manufacturer for each product of a given product type Defined in Manufacturer Specific Command Class	Product ID: Y
Location	loc	Provides location information where available.	N	(none)		
Location Name	locn	Human friendly name for location For example, "Living Room".	N	(none)		
Currency	c	Indicates the currency that is used for any monetary transactions	N	(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)		
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 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.	N	Translate Product ID to Human friendly name based upon the Product ID/product name table within Z-Wave Controller	Product ID: a unique ID identifying the actual product as defined by the manufacturer for each product of a given product type Defined in Manufacturer Specific Command Class	Product ID: Y

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	Language	Specify the language settings on a device Defined in Language Command Class	N
------------------	----	---	---	----------	---	---

365 Table 6 shows how OCF Platform properties, as specified in ISO/IEC 30118-1:2018, shall be
366 derived, typically from fields of Command Parameter of Z-Wave Command Class specified in the
367 Z-Wave Plus Device and Command Class Type Specification

368 **Table 6 – "oic.wk.p" Resource Type definition**

To OCF Property title	OCF Property name	OCF Description	OCF Mandatory	From Z-Wave Field name	Z-Wave Description	Z-Wave 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 random-generated UIUID as specified in the section 4.4 of IETF RFC 4122.	
Manufacturer Name	mnmn	Name of manufacturer (not to exceed 16 characters)	Y	Translate Manufacturer ID to Human friendly name based upon the Manufacturer ID/Manufacturer name table within Z-Wave Controller	Manufacturer ID: the unique ID identifying the manufacturer of the device. Defined in Manufacturer Specific Command Class	Y
Manufacturer Details Link (URL)	mnml	URL to manufacturer (not to exceed 32 characters)	N	(none)		
Model Number	mnmo	Model number as designated by manufacturer	N	Product ID	A unique ID identifying the actual product as defined by the manufacturer for each product of a given product type Defined in Manufacturer Specific Command Class	Y
Date of Manufacture	mndt	Manufacturing date of device	N	(none)		
Platform Version	mnpv	Version of platform – string (defined by manufacturer)	N	(none)		
OS Version	mnos	Version of platform resident OS – string (defined by manufacturer)	N	(none)		

Hardware Version	mnhw	Version of platform hardware	N	Hardware Version	A value which is unique to this particular version of the product Defined in Version Command Class	Y
Firmware version	mnfv	Version of device firmware	N	Firmware 0 Version	Dedicated to the Z-Wave chip firmware as defined by the manufacturer which assigns a version number Defined in Version Command Class	N
Support link	mnsf	URI that points to support information from manufacturer	N	(none)		
SystemTime	st	Reference time for the device	N	(none)		
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)		

369 **6.2.2.4 On-the-fly Translation**

370 If a Z-Wave Plus device is not listed in the well-defined set, a Z-Wave Bridging Function shall not
371 translate it.

372 **7 Device Type Mapping**

373 **7.1 Introduction**

374 This clause contains the mappings to Device Types.

375 **7.2 Z-Wave Device Types to OCF Device Types**

376 Table 7 captures the mapping between Z-Wave Plus defined Device Types (see Z-Wave Plus v2
377 Device Type Specification) and OCF defined Device Types.

378 **Table 7 – Z-Wave to OCF Device Type Mapping**

Classification of Z-Wave Generic Type	Z-Wave Device Type	Z-Wave Device Type ID	OCF Device Type
Multilevel Switch	Light Dimmer Switch	0x01	oic.d.light
Entry Control	DoorLock - Keypad	0x03	oic.d.smartlock
Binary Switch	On/Off Power Switch	0x01	oic.d.switch
Multilevel Sensor	Sensor - Multilevel	0x01	oic.d.sensor
Notification Sensor	Sensor - Notification	0x01	oic.d.sensor

379 Z-Wave Plus v2 device types are equivalently mapped to the Z-Wave Plus device types as
380 specified in the Z-Wave Plus v2 Device Type Specification.

381 **8 Resource to Command Class Mapping**

382 **8.1 Introduction**

383 This clause lists the set of applicable Z-Wave Command Classes and provides the OCF Resource
 384 Type(s) to which the Command Classes map along an introduction the semantics of the mapping.
 385 The detailed mappings are provided in clause 9.

386 **8.2 Z-Wave Command Classes to OCF Resources**

387 **8.2.1 Introduction**

388 This clause details the mapping between Z-Wave Command Classes and OCF defined Resource
 389 Types. Detailed Property by Property mappings are provided in clause 9.

390 Table 8 captures the mappings for Command Classes for a Z-Wave Device.

391 **Table 8 – Z-Wave Command Class to OCF Resource Type Mapping**

Z- Wave Plus Device	Z-Wave Command Class	OCF Resource Type	OCF Device Type	OCF Device Name
Light Dimmer Switch	Multilevel Switch Command Class	oic.r.switch.binary	oic.d.light	Light
	Multilevel Switch Command Class	oic.r.light.dimming		
	Manufacturer Specific Command Class Version Command Class Z-Wave Plus Info Command Class	oic.wk.d		
		oic.wk.p		
Door Lock – Keypad	Door Lock Command Class	oic.r.lock.status	oic.d.smartlock	Smart Lock
	User Code Command Class	oic.r.lock.code		
	Battery Command Class	oic.r.energy.battery.		
	Manufacturer Specific Command Class Version Command Class Z-Wave Plus Info Command Class	oic.wk.d		
		oic.wk.p		
On/Off Power Switch	Binary Switch Command Class	oic.r.switch.binary	oic.d.switch	Switch
	Battery Command Class	oic.r.energy.battery.		
	Manufacturer Specific Command Class Version Command Class Z-Wave Plus Info Command Class	oic.wk.d		
		oic.wk.p		
Sensor - Multilevel	Multilevel Sensor Command Class	oic.r.sensor.carbondioxide	oic.d.sensor	Generic Sensor
	Multilevel Sensor Command Class	oic.r.sensor.carbonmonoxide		
	Multilevel Sensor Command Class	oic.r.sensor.water		
	Multilevel Sensor Command Class	oic.r.sensor.smoke		

	Battery Command Class	oic.r.energy.battery.		
	Manufacturer Specific Command Class	oic.wk.d		
	Version Command Class	oic.wk.p		
	Z-Wave Plus Info Command Class			
Sensor - Notification	Notification Command Class	oic.r.sensor.carbondioxide	oic.d.sensor	Generic Sensor
	Notification Command Class	oic.r.sensor.carbonmonoxide		
	Notification Command Class	oic.r.sensor.water		
	Notification Command Class	oic.r.sensor.smoke		
	Battery Command Class	oic.r.energy.battery.		
	Manufacturer Specific Command Class	oic.wk.d		
Version Command Class	oic.wk.p			
	Z-Wave Plus Info Command Class			

392

393 8.2.2 Battery Command Class Mapping

394 This API defines the mapping between an instance of a Battery Command Class and the OCF
395 Battery Energy Resource. Note that the setting of the Value of OCF

396 Battery Energy to "charge" is handled via the "Battery Level" of Battery Command Class. A
397 RETRIEVE on a Batter Energy maps to Battery Get Command on an instance of a Z-Wave Battery
398 Command Class.

399

400 8.2.3 Binary Switch Command Class Mapping

401 This API defines the mapping between an instance of a Z-Wave Binary Switch Command Class
402 and an OCF Binary Switch Resource. Note that the setting of the Value of OCF Binary Switch to
403 "0x00" (off) and "0x255" (on) is handled via the "Value" of Binary Switch Command Class. A
404 RETRIEVE on a Binary Switch maps to Binary Switch Get Command on an instance of a Z-Wave
405 Binary Switch Command Class. And a UPDATE on a Binary Switch maps to Binary Switch Set
406 Command on an instance of a Z-Wave Binary Switch Command Class.

407 8.2.4 Door Lock Command Class Mapping

408 This API defines the mapping between an instance of a Door Lock Command Class and the OCF
409 Door Resource. Note that the setting of the Value of OCF Lock Status is handled via the "Value"
410 "Door Unsecured"(0x00) and "Door Secured"(0xFF) of Door Lock Command Class. A RETRIEVE
411 on a Door maps to Door Lock Operation Get Command on an instance of a Z-Wave Door Lock
412 Command Class. And a UPDATE on a Door maps to Door Lock Operation Set Command on an
413 instance of a Z-Wave Door Lock Command Class.

414 8.2.5 Multilevel Sensor Command Class Mapping

415 8.2.5.1 Multilevel Sensor Command Class Mapping for Carbon Dioxide Sensor

416 This API defines the mapping between an instance of a Z-Wave Multilevel Sensor Command Class
417 and an OCF Carbon Dioxide sensor resource. Multilevel Sensor Command Class has 5 properties:
418 Sensor Type, Precision, Scale, Size, and Sensor Value. In case Sensor Type is a carbon dioxide
419 sensor, an OCF Carbon Dioxide sensor resource is mapped. A RETRIEVE on a Carbon Dioxide

420 sensor maps to Multilevel Sensor Get Command on an instance of a Z-Wave Multilevel Sensor
421 Command Class.

422 **8.2.5.2 Multilevel Sensor Command Class Mapping for Carbon Monoxide Sensor**

423 This API defines the mapping between an instance of a Z-Wave Multilevel Sensor Command Class
424 and an OCF Carbon Monoxide sensor resource. Multilevel Sensor Command Class has 5
425 properties: Sensor Type, Precision, Scale, Size, and Sensor Value. In case Sensor Type is a
426 carbon monoxide sensor, an OCF Carbon Monoxide sensor resource is mapped. A RETRIEVE on
427 a Carbon Monoxide sensor maps to Multilevel Sensor Get Command on an instance of a Z-Wave
428 Multilevel Sensor Command Class.

429 **8.2.5.3 Multilevel Sensor Command Class Mapping for Smoke Density Sensor**

430 This API defines the mapping between an instance of a Z-Wave Multilevel Sensor Command Class
431 and an OCF Smoke sensor resource. Multilevel Sensor Command Class has 5 properties: Sensor
432 Type, Precision, Scale, Size, and Sensor Value. In case Sensor Type is a smoke density sensor,
433 an OCF Smoke sensor resource is mapped. A RETRIEVE on a Smoke sensor maps to Multilevel
434 Sensor Get Command on an instance of a Z-Wave Multilevel Sensor Command Class.

435 **8.2.5.4 Multilevel Sensor Command Class Mapping for Water Flow Sensor**

436 This API defines the mapping between an instance of a Z-Wave Multilevel Sensor Command Class
437 and an OCF Water sensor resource. Multilevel Sensor Command Class has 5 properties: Sensor
438 Type, Precision, Scale, Size, and Sensor Value. In case Sensor Type is a water flow sensor, an
439 OCF Water sensor resource is mapped. A RETRIEVE on a water sensor maps to Multilevel Sensor
440 Get Command on an instance of a Z-Wave Multilevel Sensor Command Class.

441 **8.2.6 Multilevel Switch Command Class Mapping**

442 This API defines the mapping between an instance of a Z-Wave Multilevel Switch Command Class
443 and an OCF Binary Switch Resource or an OCF Dimming Light Resource depending on the "Value"
444 of Multilevel Switch Set Command of Multilevel Switch Command Class. Note that the setting of
445 the Value of OCF Binary Switch to "0x00" (off) and "0x63" (on) and the Value of OCF Dimming
446 Light to 1 (min) and 99 (max) is handled via the "Value" of Multilevel Switch Set. A RETRIEVE on
447 a Binary Switch or Dimming Light maps to Multilevel Switch Get Command on an instance of a Z-
448 Wave Multilevel Switch Command Class. And a UPDATE on a Binary Switch or Dimming Light
449 maps to Multilevel Switch Set Command on an instance of a Z-Wave Multilevel Switch Command
450 Class.

451 **8.2.7 Notification Command Class Mapping**

452 This API defines the mapping between an instance of a Notification Command Class and OCF
453 Specific sensor resources. Notification Command Class has 9 properties; these map as follows:
454 V1 Alarm Type, V1 Alarm Level, Notification Status, Notification Type, Notification Event:State,
455 Sequence, Event:State Parameters Length, Event:State Parameter, Sequence Number =>
456 corresponding properties of smoke sensor, carbon monoxide sensor, carbon dioxide sensor or
457 water sensor. This is presented in OCF as the distinct Resource instances. A RETRIEVE on a
458 Specific Sensor maps to Notification Get Command on an instance of a Z-Wave Notification
459 Command Class. And a UPDATE on a Specific Sensor maps to Notification Set Command on an
460 instance of a Z-Wave Notification Command Class.

461 **8.2.8 User Code Command Class Mapping**

462 This API defines the mapping between an instance of a User Code Command Class and the OCF
463 Lock Code Resource. A RETRIEVE on a Lock Code maps to User Code Get Command on an
464 instance of a Z-Wave User Code Command Class. And a UPDATE on a Lock Code maps to User
465 Code Set Command on an instance of a Z-Wave User Code Command Class.

466 **9 Detailed Mapping APIs**

467 This clause provides a mapping description (using JSON that aligns with the Derived Modelling
 468 syntax described in Derived Models for Interoperability between IoT Ecosystems) for all Command
 469 Classes and Resources that are within scope.

470 Table 9 provides a reference and link to the per Command Class clauses.

471 **Table 9 – Command Class to Resource Summary**

Z-Wave Command Class Name	Mapped Resource(s)	Mapping Clause
Battery Command Class	oic.r.energy.battery	8.2.2
Binary Switch Command Class	oic.r.switch.binary	8.2.3
Door Lock Command Class	oic.r.lock.status	8.2.4
Multilevel Sensor Command Class	oic.r.sensor.carbondioxide	8.2.5.1
Multilevel Sensor Command Class	oic.r.sensor.carbonmonoxide	8.2.5.2
Multilevel Sensor Command Class	oic.r.sensor.water	8.2.5.4
Multilevel Sensor Command Class	oic.r.sensor.smoke	8.2.5.3
Multilevel Switch Command Class	oic.r.switch.binary	8.2.6
	oic.r.light.dimming	
Notification Command Class	oic.r.sensor.carbondioxide	8.2.7
Notification Command Class	oic.r.sensor.carbonmonoxide	
Notification Command Class	oic.r.sensor.water	
Notification Command Class	oic.r.sensor.smoke	
User Code Command Class	oic.r.lock.status	8.2.8

472

473 **9.1 Battery Command Class**

474 **9.1.1 Derived model**

475 The derived model: "zwave.operation.batterycommandclass".

476 **9.1.2 Property definition**

477 Table 10 provides the detailed per Property mapping for "zwave.operation.batterycommandclass".

478 **Table 10 – The Property mapping for "zwave.operation.batterycommandclass".**

Z-Wave Property name	OCF Resource	To OCF	From OCF
Battery Level	oic.r.energy.battery	if Battery Level = 255, ocf.r.energy.battery.lowbattery = true; ocf.r.energy.battery.charge = 0. if Battery Level != 255, ocf.r.energy.battery.charge = Battery Level.	N/A

479 Table 11 provides the details of the Properties that are part of
 480 "zwave.operation.batterycommandclass".

Table 11 – The Properties of "zwave.operation.batterycommandclass".

Z-Wave Property name	Type	Required	Description
Battery Level	if Battery Level = 255, string if Battery Level != 255, integer	yes	percentage indicating the battery level or low battery warning

482 9.1.3 Derived model definition

```

483 {
484   "id":
485   "http://openinterconnect.org/zwavemapping/schemas/zwave.operation.batterycommandclass.json#",
486   "$schema": "http://json-schema.org/draft-04/schema#",
487   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
488   "title": "Battery Command Class",
489   "definitions": {
490     "zwave.operation.batterycommandclass": {
491       "type": "object",
492       "properties": {
493         "Battery Level": {
494           "type": [
495             "if Battery Level = 255, string",
496             "if Battery Level != 255, integer"
497           ],
498           "description": "percentage indicating the battery level or low battery warning",
499           "x-ocf-conversion": {
500             "x-ocf-alias": "oic.r.energy.battery",
501             "x-to-ocf": [
502               "if Battery Level = 255, ocf.r.energy.battery.lowbattery = true;
503               ocf.r.energy.battery.charge = 0.",
504               "if Battery Level != 255, ocf.r.energy.battery.charge = Battery Level."
505             ],
506             "x-from-ocf": [
507               "N/A"
508             ]
509           }
510         }
511       }
512     }
513   },
514   "type": "object",
515   "allOf": [
516     {"$ref": "#/definitions/zwave.operation.batterycommandclass"}
517   ],
518   "required": ["Battery Level"]
519 }

```

520 9.2 Binary Switch Command Classes

521 9.2.1 Derived model

522 The derived model: "zwave.operation.binaryswitchcommandclass".

523 9.2.2 Property definition

524 Table 12 provides the detailed per Property mapping for
525 "zwave.operation.binaryswitchcommandclass".

526 **Table 12 – The Property mapping for "zwave.operation.binaryswitchcommandclass".**

Z-Wave Property name	OCF Resource	To OCF	From OCF
Value	oic.r.switch.binary	if Value = 255, ocf.r.switch.binary.value = true. if Value != 255, ocf.r.switch.binary.value = false.	if ocf.r.switch.binary.value = false, Value = 0 if ocf.r.switch.binary.value = true, Value = 255

527 Table 13 provides the details of the Properties that are part of
 528 "zwave.operation.binaryswitchcommandclass".

529 **Table 13 – The Properties of "zwave.operation.binaryswitchcommandclass".**

Z-Wave Property name	Type	Required	Description
Value	boolean	yes	On/Off state at the receiving node

530 **9.2.3 Derived model definition**

```

531 {
532   "id":
533   "http://openinterconnect.org/zwavemapping/schemas/zwave.operation.binaryswitchcommandclass.json#",
534   "$schema": "http://json-schema.org/draft-04/schema#",
535   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
536   "title": "Binary Switch Command Class",
537   "definitions": {
538     "zwave.operation.binaryswitchcommandclass": {
539       "type": "object",
540       "properties": {
541         "Value": {
542           "type": "boolean",
543           "description": "On/Off state at the receiving node",
544           "x-ocf-conversion": {
545             "x-ocf-alias": "oic.r.switch.binary",
546             "x-to-ocf": [
547               "if Value = 255, ocf.r.switch.binary.value = true.",
548               "if Value != 255, ocf.r.switch.binary.value = false."
549             ],
550             "x-from-ocf": [
551               "if ocf.r.switch.binary.value = false, Value = 0",
552               "if ocf.r.switch.binary.value = true, Value = 255"
553             ]
554           }
555         }
556       }
557     }
558   },
559   "type": "object",
560   "allOf": [
561     {"$ref": "#/definitions/zwave.operation.binaryswitchcommandclass"}
562   ],
563   "required": ["Value"]
564 }
565 
```

566 **9.3 Door Lock Command Class**

567 **9.3.1 Derived model**

568 The derived model: "zwave.operation.doorlockcommandclass".

569 **9.3.2 Property definition**

570 Table 14 provides the detailed per Property mapping for "zwave.operation.doorlockcommandclass".

571 **Table 14 – The Property mapping for "zwave.operation.doorlockcommandclass".**

Z-Wave Property name	OCF Resource	To OCF	From OCF
Door Lock Mode	oic.r.lock.status	if Door Lock Mode = 0x00, ocf.r.lock.status.lockState = UnLocked if Door Lock Mode = 0xFF, ocf.r.lock.status.lockState = Locked	if ocf.r.lock.status.lockState = UnLocked, Door Lock Mode = 0x00 if ocf.r.lock.status.lockState = Locked, Door Lock Mode = 0xFF

572 Table 15 provides the details of the Properties that are part of
 573 "zwave.operation.doorlockcommandclass".

574 **Table 15 – The Properties of "zwave.operation.doorlockcommandclass".**

Z-Wave Property name	Type	Required	Description
Door Lock Mode	integer	yes	operation mode of the door lock device

575 **9.3.3 Derived model definition**

```

576 {
577   "id":
578   "http://openinterconnect.org/zwavemapping/schemas/zwave.operation.doorlockcommandclass.json#",
579   "$schema": "http://json-schema.org/draft-04/schema#",
580   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
581   "title": "Door Lock Command Class",
582   "definitions": {
583     "zwave.operation.doorlockcommandclass": {
584       "type": "object",
585       "properties": {
586         "Door Lock Mode": {
587           "type": "integer",
588           "description": "operation mode of the door lock device",
589           "x-ocf-conversion": {
590             "x-ocf-alias": "oic.r.lock.status",
591             "x-to-ocf": [
592               "if Door Lock Mode = 0x00, ocf.r.lock.status.lockState = UnLocked",
593               "if Door Lock Mode = 0xFF, ocf.r.lock.status.lockState = Locked"
594             ],
595             "x-from-ocf": [
596               "if ocf.r.lock.status.lockState = Unlocked, Door Lock Mode = 0x00",
597               "if ocf.r.lock.status.lockState = Locked, Door Lock Mode = 0xFF"
598             ]
599           }
600         }
601       }
602     },
603     "type": "object",
604     "allOf": [
605       {"$ref": "#/definitions/zwave.operation.doorlockcommandclass"}
606     ],
607     "required": ["Door Lock Mode"]
608   }
609 }

```

610 **9.4 Multilevel Sensor Command Class Carbon Dioxide**

611 **9.4.1 Derived model**

612 The derived model: "zwave.operation.multilevelsensorcommandclasscarbondioxide".

613 **9.4.2 Property definition**

614 Table 16 provides the detailed per Property mapping for
 615 "zwave.operation.multilevelsensorcommandclasscarbondioxide".

616 **Table 16 – The Property mapping for**
 617 **"zwave.operation.multilevelsensorcommandclasscarbondioxide".**

Z-Wave Property name	OCF Resource	To OCF	From OCF
Sensor Type	oic.r.sensor.carbondioxide	if Sensor Type = Carbon dioxide CO2-level, ocf.rt = oic.r.sensor.carbondioxide.	N/A
Precision	oic.r.sensor.carbondioxide	ocf.r.sensor.carbondioxide.precision = Precision	N/A

Scale	oic.r.sensor.carbondioxide	N/A	Scale = ppm (0x00)
Size	oic.r.sensor.carbondioxide	N/A	N/A
Sensor Value	oic.r.sensor.carbondioxide	ocf.r.sensor.carbondioxide.value = trueocf.r.sensor.carbondioxide.measurement = Sensor Value	N/A

618 Table 17 provides the details of the Properties that are part of
619 "zwave.operation.multilevelsensorcommandclasscarbondioxide".

620
621

**Table 17 – The Properties of
"zwave.operation.multilevelsensorcommandclasscarbondioxide".**

Z-Wave Property name	Type	Required	Description
Sensor Type	Integer	yes	specify the carbon dioxide sensor type of the actual sensor reading
Precision	Number	yes	indicate how many decimal places are included the Sensor Value field
Scale	Integer	yes	indicate what scale is used for the actual sensor reading
Size	enum	yes	indicate the length in bytes of the Sensor Value field
Sensor Value	array	yes	specify the value of the actual sensor reading

622 **9.4.3 Derived model definition**

```

623 {
624   "id":
625   "http://openinterconnect.org/zwavemapping/schemas/zwave.operation.multilevelsensorcommandclasscarbo
626   ndioxide.json#",
627   "$schema": "http://json-schema.org/draft-04/schema#",
628   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
629   "title": "Multilevel Sensor Command Class Carbon Dioxide",
630   "definitions": {
631     "zwave.operation.multilevelsensorcommandclasscarbondioxide": {
632       "type": "object",
633       "properties": {
634         "Sensor Type": {
635           "type": "Integer",
636           "description": "specify the carbon dioxide sensor type of the actual sensor reading ",
637           "x-ocf-conversion": {
638             "x-ocf-alias": "oic.r.sensor.carbondioxide",
639             "x-to-ocf": [
640               "if Sensor Type = Carbon dioxide CO2-level, ocf.rt = oic.r.sensor.carbondioxide."
641             ],
642             "x-from-ocf": [
643               "N/A"
644             ]
645           }
646         },
647         "Precision": {
648           "type": "Number",
649           "description": "indicate how many decimal places are included the Sensor Value field ",
650           "x-ocf-conversion": {
651             "x-ocf-alias": "oic.r.sensor.carbondioxide",
652             "x-to-ocf": [
653               "ocf.r.sensor.carbondioxide.precision = Precision"
654             ],

```

```

655         "x-from-ocf": [
656             "N/A"
657         ]
658     },
659 },
660 "Scale": {
661     "type": "Integer",
662     "description": " indicate what scale is used for the actual sensor reading ",
663     "x-ocf-conversion": {
664         "x-ocf-alias": "oic.r.sensor.carbondioxide",
665         "x-to-ocf": [
666             "N/A"
667         ],
668         "x-from-ocf": [
669             "Scale = ppm (0x00)"
670         ]
671     }
672 },
673 "Size": {
674     "type": "enum",
675     "description": " indicate the length in bytes of the Sensor Value field ",
676     "x-ocf-conversion": {
677         "x-ocf-alias": "oic.r.sensor.carbondioxide",
678         "x-to-ocf": [
679             "N/A"
680         ],
681         "x-from-ocf": [
682             "N/A"
683         ]
684     }
685 },
686 "Sensor Value": {
687     "type": "array",
688     "description": " specify the value of the actual sensor reading ",
689     "x-ocf-conversion": {
690         "x-ocf-alias": "oic.r.sensor.carbondioxide",
691         "x-to-ocf": [
692             "ocf.r.sensor.carbondioxide.value = true",
693             "ocf.r.sensor.carbondioxide.measurement = Sensor Value"
694         ],
695         "x-from-ocf": [
696             "N/A"
697         ]
698     }
699 }
700 }
701 }
702 },
703 "type": "object",
704 "allOf": [
705     {"$ref": "#/definitions/zwave.operation.multilevelsensorcommandclasscarbondioxide"}
706 ],
707 "required": ["Sensor Type", "Precision", "Scale", "Size", "Sensor Value"]
708 }
709

```

710 9.5 Multilevel Sensor Command Class Carbon Monoxide

711 9.5.1 Derived model

712 The derived model: "zwave.operation.multilevelsensorcommandclasscarbonmonoxide".

713 9.5.2 Property definition

714 Table 18 provides the detailed per Property mapping for
715 "zwave.operation.multilevelsensorcommandclasscarbonmonoxide".

716
717

Table 18 – The Property mapping for "zwave.operation.multilevelsensorcommandclasscarbonmonoxide".

Z-Wave Property name	OCF Resource	To OCF	From OCF
Sensor Type	oic.r.sensor.carbonmonoxide	if Sensor Type = Carbon monoxide (CO) level, ocf.rt = oic.r.sensor.carbonmonoxide.	N/A
Precision	oic.r.sensor.carbonmonoxide	ocf.r.sensor.carbonmonoxide.precision = Precision	N/A
Scale	oic.r.sensor.carbonmonoxide	N/A	Scale = ppm (0x01)
Size	oic.r.sensor.carbonmonoxide	N/A	N/A
Sensor Value	oic.r.sensor.carbonmonoxide	ocf.r.sensor.carbonmonoxide.value = trueocf.r.sensor.carbonmonoxide.measurement = Sensor Value	N/A

718 Table 19 provides the details of the Properties that are part of
719 "zwave.operation.multilevelsensorcommandclasscarbonmonoxide".

720
721

Table 19 – The Properties of "zwave.operation.multilevelsensorcommandclasscarbonmonoxide".

Z-Wave Property name	Type	Required	Description
Sensor Type	Integer	yes	specify the carbon monoxidesensor type of the actual sensor reading
Precision	Number	yes	indicate how many decimal places are included the Sensor Value field
Scale	Integer	yes	indicate what scale is used for the actual sensor reading
Size	enum	yes	indicate the length in bytes of the Sensor Value field
Sensor Value	array	yes	specify the value of the actual sensor reading

722 **9.5.3 Derived model definition**

```
723 {
724   "id":
725   "http://openinterconnect.org/zwavemapping/schemas/zwave.operation.multilevelsensorcommandclasscarbo
726   nmonoxide.json#",
727   "$schema": "http://json-schema.org/draft-04/schema#",
728   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
729   "title": "Multilevel Sensor Command Class Carbon Monoxide",
730   "definitions": {
731     "zwave.operation.multilevelsensorcommandclasscarbonmonoxide": {
732       "type": "object",
733       "properties": {
734         "Sensor Type": {
735           "type": "Integer",
736           "description": "specify the carbon monoxidesensor type of the actual sensor reading",
737           "x-ocf-conversion": {
738             "x-ocf-alias": "oic.r.sensor.carbonmonoxide",
739             "x-to-ocf": [
740               "if Sensor Type = Carbon monoxide (CO) level, ocf.rt = oic.r.sensor.carbonmonoxide."
741             ],
742             "x-from-ocf": [
```

```

743         "N/A"
744     ]
745 }
746 },
747 "Precision": {
748     "type" : "Number",
749     "description": " indicate how many decimal places are included the Sensor Value field ",
750     "x-ocf-conversion": {
751         "x-ocf-alias": "oic.r.sensor.carbonmonoxide",
752         "x-to-ocf": [
753             "ocf.r.sensor.carbonmonoxide.precision = Precision"
754         ],
755         "x-from-ocf": [
756             "N/A"
757         ]
758     }
759 },
760 "Scale": {
761     "type" : "Integer",
762     "description": " indicate what scale is used for the actual sensor reading ",
763     "x-ocf-conversion": {
764         "x-ocf-alias": "oic.r.sensor.carbonmonoxide",
765         "x-to-ocf": [
766             "N/A"
767         ],
768         "x-from-ocf": [
769             "Scale = ppm (0x01)"
770         ]
771     }
772 },
773 "Size": {
774     "type" : "enum",
775     "description": " indicate the length in bytes of the Sensor Value field ",
776     "x-ocf-conversion": {
777         "x-ocf-alias": "oic.r.sensor.carbonmonoxide",
778         "x-to-ocf": [
779             "N/A"
780         ],
781         "x-from-ocf": [
782             "N/A"
783         ]
784     }
785 },
786 "Sensor Value": {
787     "type" : "array",
788     "description": " specify the value of the actual sensor reading ",
789     "x-ocf-conversion": {
790         "x-ocf-alias": "oic.r.sensor.carbonmonoxide",
791         "x-to-ocf": [
792             "ocf.r.sensor.carbonmonoxide.value = true",
793             "ocf.r.sensor.carbonmonoxide.measurement = Sensor Value"
794         ],
795         "x-from-ocf": [
796             "N/A"
797         ]
798     }
799 }
800 }
801 }
802 },
803 "type": "object",
804 "allOf": [
805     {"$ref": "#/definitions/zwave.operation.multilevelsensorcommandclasscarbonmonoxide"}
806 ],
807 "required": ["Sensor Type", "Precision", "Scale", "Size", "Sensor Value"]
808 }
809

```

810 **9.6 Multilevel Sensor Command Class Smoke Density**

811 **9.6.1 Derived model**

812 The derived model: "zwave.operation.multilevelsensorcommandclasssmokedensity".

813 **9.6.2 Property definition**

814 Table 20 provides the detailed per Property mapping for
815 "zwave.operation.multilevelsensorcommandclasssmokedensity".

816 **Table 20 – The Property mapping for**
817 **"zwave.operation.multilevelsensorcommandclasssmokedensity".**

Z-Wave Property name	OCF Resource	To OCF	From OCF
Sensor Type	oic.r.sensor.smoke	if Sensor Type = Smoke density, ocf.rt = oic.r.sensor.smoke.	N/A
Precision	oic.r.sensor.smoke	ocf.r.sensor.smoke.precision = Precision	N/A
Scale	oic.r.sensor.smoke	N/A	Scale = percent (0x00)
Size	oic.r.sensor.smoke	N/A	N/A
Sensor Value	oic.r.sensor.smoke	ocf.r.sensor.smoke.value = trueocf.r.sensor.smoke.measurement = Sensor Value	N/A

818 Table 21 provides the details of the Properties that are part of
819 "zwave.operation.multilevelsensorcommandclasssmokedensity".

820 **Table 21 – The Properties of**
821 **"zwave.operation.multilevelsensorcommandclasssmokedensity".**

Z-Wave Property name	Type	Required	Description
Sensor Type	Integer	yes	specify the smoke density sensor type of the actual sensor reading
Precision	Number	yes	indicate how many decimal places are included the Sensor Value field
Scale	Integer	yes	indicate what scale is used for the actual sensor reading
Size	enum	yes	indicate the length in bytes of the Sensor Value field
Sensor Value	array	yes	specify the value of the actual sensor reading

822 **9.6.3 Derived model definition**

```
823 {
824   "id":
825   "http://openinterconnect.org/zwavemapping/schemas/zwave.operation.multilevelsensorcommandclasssmoke
826   density.json#",
827   "$schema": "http://json-schema.org/draft-04/schema#",
828   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
829   "title": "Multilevel Sensor Command Class Smoke Density",
830   "definitions": {
831     "zwave.operation.multilevelsensorcommandclasssmokedensity": {
```

```

832 "type": "object",
833 "properties": {
834   "Sensor Type": {
835     "type" : "Integer",
836     "description": " specify the smoke density sensor type of the actual sensor reading ",
837     "x-ocf-conversion": {
838       "x-ocf-alias": "oic.r.sensor.smoke",
839       "x-to-ocf": [
840         "if Sensor Type = Smoke density, ocf.rt = oic.r.sensor.smoke."
841       ],
842       "x-from-ocf": [
843         "N/A"
844       ]
845     }
846   },
847   "Precision": {
848     "type" : "Number",
849     "description": " indicate how many decimal places are included the Sensor Value field ",
850     "x-ocf-conversion": {
851       "x-ocf-alias": "oic.r.sensor.smoke",
852       "x-to-ocf": [
853         "ocf.r.sensor.smoke.precision = Precision"
854       ],
855       "x-from-ocf": [
856         "N/A"
857       ]
858     }
859   },
860   "Scale": {
861     "type" : "Integer",
862     "description": " indicate what scale is used for the actual sensor reading ",
863     "x-ocf-conversion": {
864       "x-ocf-alias": "oic.r.sensor.smoke",
865       "x-to-ocf": [
866         "N/A"
867       ],
868       "x-from-ocf": [
869         "Scale = percent (0x00)"
870       ]
871     }
872   },
873   "Size": {
874     "type" : "enum",
875     "description": " indicate the length in bytes of the Sensor Value field ",
876     "x-ocf-conversion": {
877       "x-ocf-alias": "oic.r.sensor.smoke",
878       "x-to-ocf": [
879         "N/A"
880       ],
881       "x-from-ocf": [
882         "N/A"
883       ]
884     }
885   },
886   "Sensor Value": {
887     "type" : "array",
888     "description": " specify the value of the actual sensor reading ",
889     "x-ocf-conversion": {
890       "x-ocf-alias": "oic.r.sensor.smoke",
891       "x-to-ocf": [
892         "ocf.r.sensor.smoke.value = true",
893         "ocf.r.sensor.smoke.measurement = Sensor Value"
894       ],
895       "x-from-ocf": [
896         "N/A"
897       ]
898     }
899   }
900 }
901 },
902 },

```

```

903     "type": "object",
904     "allOf": [
905       {"$ref": "#/definitions/zwave.operation.multilevelsensorcommandclasssmokedensity"}
906     ],
907     "required": ["Sensor Type", "Precision", "Scale", "Size", "Sensor Value"]
908   }
909 }

```

9.7 Multilevel Sensor Command Class Water Flow

9.7.1 Derived model

The derived model: "zwave.operation.multilevelsensorcommandclasswaterflow".

9.7.2 Property definition

Table 22 provides the detailed per Property mapping for "zwave.operation.multilevelsensorcommandclasswaterflow".

916 **Table 22 – The Property mapping for**
 917 **"zwave.operation.multilevelsensorcommandclasswaterflow".**

Z-Wave Property name	OCF Resource	To OCF	From OCF
Sensor Type	oic.r.sensor.water	if Sensor Type = Water flow, ocf.rt = oic.r.sensor.water.	N/A
Precision	oic.r.sensor.water	ocf.r.sensor.water.precision = Precision	N/A
Scale	oic.r.sensor.water	N/A	Scale = litre/hr (0x00)
Size	oic.r.sensor.water	N/A	N/A
Sensor Value	oic.r.sensor.water	ocf.r.sensor.water.value = trueocf.r.sensor.water.measurement = Sensor Value	N/A

918 Table 23 provides the details of the Properties that are part of
 919 "zwave.operation.multilevelsensorcommandclasswaterflow".

920 **Table 23 – The Properties of "zwave.operation.multilevelsensorcommandclasswaterflow".**

Z-Wave Property name	Type	Required	Description
Sensor Type	Integer	yes	specify the water flow sensor type of the actual sensor reading
Precision	Number	yes	indicate how many decimal places are included the Sensor Value field
Scale	Integer	yes	indicate what scale is used for the actual sensor reading
Size	enum	yes	indicate the length in bytes of the Sensor Value field
Sensor Value	array	yes	specify the value of the actual sensor reading

9.7.3 Derived model definition

```

922 {
923   "id":
924   "http://openinterconnect.org/zwavemapping/schemas/zwave.operation.multilevelsensorcommandclasswater

```



```

925 flow.json#",
926 "$schema": "http://json-schema.org/draft-04/schema#",
927 "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
928 "title": "Multilevel Sensor Command Class Water Flow",
929 "definitions": {
930   "zwave.operation.multilevelsensorcommandclasswaterflow": {
931     "type": "object",
932     "properties": {
933       "Sensor Type": {
934         "type": "Integer",
935         "description": " specify the water flow sensor type of the actual sensor reading ",
936         "x-ocf-conversion": {
937           "x-ocf-alias": "oic.r.sensor.water",
938           "x-to-ocf": [
939             "if Sensor Type = Water flow, ocf.rt = oic.r.sensor.water."
940           ],
941           "x-from-ocf": [
942             "N/A"
943           ]
944         }
945       },
946       "Precision": {
947         "type": "Number",
948         "description": " indicate how many decimal places are included the Sensor Value field ",
949         "x-ocf-conversion": {
950           "x-ocf-alias": "oic.r.sensor.water",
951           "x-to-ocf": [
952             "ocf.r.sensor.water.precision = Precision"
953           ],
954           "x-from-ocf": [
955             "N/A"
956           ]
957         }
958       },
959       "Scale": {
960         "type": "Integer",
961         "description": " indicate what scale is used for the actual sensor reading ",
962         "x-ocf-conversion": {
963           "x-ocf-alias": "oic.r.sensor.water",
964           "x-to-ocf": [
965             "N/A"
966           ],
967           "x-from-ocf": [
968             "Scale = litre/hr (0x00)"
969           ]
970         }
971       },
972       "Size": {
973         "type": "enum",
974         "description": " indicate the length in bytes of the Sensor Value field ",
975         "x-ocf-conversion": {
976           "x-ocf-alias": "oic.r.sensor.water",
977           "x-to-ocf": [
978             "N/A"
979           ],
980           "x-from-ocf": [
981             "N/A"
982           ]
983         }
984       },
985       "Sensor Value": {
986         "type": "array",
987         "description": " specify the value of the actual sensor reading ",
988         "x-ocf-conversion": {
989           "x-ocf-alias": "oic.r.sensor.water",
990           "x-to-ocf": [
991             "ocf.r.sensor.water.value = true",
992             "ocf.r.sensor.water.measurement = Sensor Value"
993           ],
994           "x-from-ocf": [
995             "N/A"

```

```

996         ]
997     }
998 }
999 }
1000 }
1001 },
1002 "type": "object",
1003 "allOf": [
1004   {"$ref": "#/definitions/zwave.operation.multilevelsensorcommandclasswaterflow"}
1005 ],
1006 "required": ["Sensor Type", "Precision", "Scale", "Size", "Sensor Value"]
1007 }
1008

```

1009 **9.8 Multilevel Switch Command Class**

1010 **9.8.1 Derived model**

1011 The derived model: "zwave.operation.multilevelswitchcommandclass".

1012 **9.8.2 Property definition**

1013 Table 24 provides the detailed per Property mapping for
 1014 "zwave.operation.multilevelswitchcommandclass".

1015 **Table 24 – The Property mapping for "zwave.operation.multilevelswitchcommandclass".**

Z-Wave Property name	OCF Resource	To OCF	From OCF
Value	oic.r.switch.binary, oic.r.light.dimming	if value = 0, ocf.rt = oic.r.switch.binary & ocf.r.switch.binary.value = falseotherwise: ocf.rt = oic.r.light.dimming; ocf.r.light.dimming.dimmingSetting = value	value = dimmingSettingif ocf.rt = oic.r.switch.binary, value = ocf.r.switch.binary.valueif ocf.rt = oic.r.light.dimming, value = dimmingSetting

1016 Table 25 provides the details of the Properties that are part of
 1017 "zwave.operation.multilevelswitchcommandclass".

1018 **Table 25 – The Properties of "zwave.operation.multilevelswitchcommandclass".**

Z-Wave Property name	Type	Required	Description
Value	integer, boolean	yes	multilevel value in a supporting device

1019 **9.8.3 Derived model definition**

```

1020 {
1021   "id":
1022   "http://openinterconnect.org/zwavemapping/schemas/zwave.operation.multilevelswtichcommandclass.json
1023   #",
1024   "$schema": "http://json-schema.org/draft-04/schema#",
1025   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1026   "title": "Multilevel Switch Command Class",
1027   "definitions": {
1028     "zwave.operation.multilevelswitchcommandclass": {
1029       "type": "object",
1030       "properties": {
1031         "Value": {
1032           "type": "integer, boolean",
1033           "description": "multilevel value in a supporting device",
1034           "x-ocf-conversion": {
1035             "x-ocf-alias": "oic.r.switch.binary, oic.r.light.dimming",
1036             "x-to-ocf": [
1037               "if value = 0, ocf.rt = oic.r.switch.binary & ocf.r.switch.binary.value = false",
1038               "otherwise: ocf.rt = oic.r.light.dimming; ocf.r.light.dimming.dimmingSetting = value"
1039             ],

```

```

1040     "x-from-ocf": [
1041         "value = dimmingSetting",
1042         "if ocf.rt = oic.r.switch.binary, value = ocf.r.switch.binary.value",
1043         "if ocf.rt = oic.r.light.dimming, value = dimmingSetting"
1044     ]
1045     }
1046   }
1047 }
1048 },
1049 },
1050 "type": "object",
1051 "allOf": [
1052   {"$ref": "#/definitions/zwave.operation.multilevelswitchcommandclass"}
1053 ],
1054 "required": ["Value"]
1055 }

```

1056 **9.9 Notification Command Class**

1057 **9.9.1 Derived model**

1058 The derived model: "zwave.operation.notificationcommandclass".

1059 **9.9.2 Property definition**

1060 Table 26 provides the detailed per Property mapping for
 1061 "zwave.operation.notificationcommandclass".

1062 **Table 26 – The Property mapping for "zwave.operation.notificationcommandclass".**

Z-Wave Property name	OCF Resource	To OCF	From OCF
V1 Alarm Type	oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide, oic.r.sensor.smoke, oic.r.sensor.water	N/A	N/A
V1 Alarm Level	oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide, oic.r.sensor.smoke, oic.r.sensor.water	N/A	N/A
Notification Status	oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide, oic.r.sensor.smoke, oic.r.sensor.water	Value = Notification Status	N/A
Notification Type	oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide, oic.r.sensor.smoke, oic.r.sensor.water	if Notification Type = Smoke Alarm, ocf.rt = oic.r.sensor.smoke.if Notification Type = CO Alarm, ocf.rt = oic.r.sensor.carbonmonoxide.if Notification Type = CO2 Alarm, ocf.rt = oic.r.sensor.carbondioxide.if Notification Type = Water Alarm, ocf.rt = oic.r.sensor.water.	N/A
Notification Event:State	oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide, oic.r.sensor.smoke, oic.r.sensor.water	Value = Notification Event:State	N/A
Sequence	oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide, oic.r.sensor.smoke, oic.r.sensor.water	ocf.sequence = Sequence	N/A

Event:State Parameters Length	oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide, oic.r.sensor.smoke, oic.r.sensor.water	ocf.event:stateparameterslength = Event:State Parameters Length	N/A
Event:State Parameter	oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide, oic.r.sensor.smoke, oic.r.sensor.water	ocf.event:stateparameter = Event:State Parameter	N/A
Sequence Number	oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide, oic.r.sensor.smoke, oic.r.sensor.water	ocf.sequencenumber = Sequence Number	N/A

1063 Table 27 provides the details of the Properties that are part of
1064 "zwave.operation.notificationcommandclass".

1065 **Table 27 – The Properties of "zwave.operation.notificationcommandclass".**

Z-Wave Property name	Type	Required	Description
V1 Alarm Type	Integer	yes	depends on the V1 Alarm field advertised in the Alarm Type Supported Report Command
V1 Alarm Level	Integer	yes	product manual specific
Notification Status	Integer	yes	advertise the status of the Notification Type
Notification Type	Integer	yes	specify a Notification Type
Notification Event:State	Integer	yes	specify a Notification Event/State for the advertised Notification Type
Sequence	boolean	yes	advertise the presence of the Sequence Number field
Event:State Parameters Length	number	yes	advertise the length in bytes of the Event / State Parameters field
Event:State Parameter	Integer	no	specify associated parameters to a Notification
Sequence Number	number	no	advertise a sequence number for the actual Notification

1066 **9.9.3 Derived model definition**

```
1067 {
1068   "id":
1069   "http://openinterconnect.org/zwavemapping/schemas/zwave.operation.notificationcommandclass.json#",
1070   "$schema": "http://json-schema.org/draft-04/schema#",
1071   "description" : "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1072   "title": "Notification Command Class",
1073   "definitions": {
1074     "zwave.operation.notificationcommandclass": {
1075       "type": "object",
1076       "properties": {
1077         "V1 Alarm Type": {
1078           "type" : "Integer",
1079           "description": "depends on the V1 Alarm field advertised in the Alarm Type Supported
1080 Report Command",
```

```

1081         "x-ocf-conversion": {
1082             "x-ocf-alias": "oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide,
1083 oic.r.sensor.smoke, oic.r.sensor.water",
1084             "x-to-ocf": [
1085                 "N/A"
1086             ],
1087             "x-from-ocf": [
1088                 "N/A"
1089             ]
1090         }
1091     },
1092     "V1 Alarm Level": {
1093         "type" : "Integer",
1094         "description": "product manual specific",
1095         "x-ocf-conversion": {
1096             "x-ocf-alias": "oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide,
1097 oic.r.sensor.smoke, oic.r.sensor.water",
1098             "x-to-ocf": [
1099                 "N/A"
1100             ],
1101             "x-from-ocf": [
1102                 "N/A"
1103             ]
1104         }
1105     },
1106     "Notification Status": {
1107         "type" : "Integer",
1108         "description": "advertise the status of the Notification Type",
1109         "x-ocf-conversion": {
1110             "x-ocf-alias": "oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide,
1111 oic.r.sensor.smoke, oic.r.sensor.water",
1112             "x-to-ocf": [
1113                 "Value = Notification Status"
1114             ],
1115             "x-from-ocf": [
1116                 "N/A"
1117             ]
1118         }
1119     },
1120     "Notification Type": {
1121         "type" : "Integer",
1122         "description": " specify a Notification Type ",
1123         "x-ocf-conversion": {
1124             "x-ocf-alias": "oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide,
1125 oic.r.sensor.smoke, oic.r.sensor.water",
1126             "x-to-ocf": [
1127                 "if Notification Type = Smoke Alarm, ocf.rt = oic.r.sensor.smoke.",
1128                 "if Notification Type = CO Alarm, ocf.rt = oic.r.sensor.carbonmonoxide.",
1129                 "if Notification Type = CO2 Alarm, ocf.rt = oic.r.sensor.carbondioxide.",
1130                 "if Notification Type = Water Alarm, ocf.rt = oic.r.sensor.water."
1131             ],
1132             "x-from-ocf": [
1133                 "N/A"
1134             ]
1135         }
1136     },
1137     "Notification Event:State": {
1138         "type" : "Integer",
1139         "description": "specify a Notification Event/State for the advertised Notification
1140 Type",
1141         "x-ocf-conversion": {
1142             "x-ocf-alias": "oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide,
1143 oic.r.sensor.smoke, oic.r.sensor.water",
1144             "x-to-ocf": [
1145                 "Value = Notification Event:State"
1146             ],
1147             "x-from-ocf": [
1148                 "N/A"
1149             ]
1150         }
1151     },

```

```

1152     "Sequence": {
1153         "type" : "boolean",
1154         "description": "advertise the presence of the Sequence Number field",
1155         "x-ocf-conversion": {
1156             "x-ocf-alias": "oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide,
1157 oic.r.sensor.smoke, oic.r.sensor.water",
1158             "x-to-ocf": [
1159                 "ocf.sequence = Sequence"
1160             ],
1161             "x-from-ocf": [
1162                 "N/A"
1163             ]
1164         }
1165     },
1166     "Event:State Parameters Length": {
1167         "type" : "number",
1168         "description": "advertise the length in bytes of the Event / State Parameters field",
1169         "x-ocf-conversion": {
1170             "x-ocf-alias": "oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide,
1171 oic.r.sensor.smoke, oic.r.sensor.water",
1172             "x-to-ocf": [
1173                 "ocf.event:stateparameterslength = Event:State Parameters Length"
1174             ],
1175             "x-from-ocf": [
1176                 "N/A"
1177             ]
1178         }
1179     },
1180     "Event:State Parameter": {
1181         "type" : "Integer",
1182         "description": "specify associated parameters to a Notification",
1183         "x-ocf-conversion": {
1184             "x-ocf-alias": "oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide,
1185 oic.r.sensor.smoke, oic.r.sensor.water",
1186             "x-to-ocf": [
1187                 "ocf.event:stateparameter = Event:State Parameter"
1188             ],
1189             "x-from-ocf": [
1190                 "N/A"
1191             ]
1192         }
1193     },
1194     "Sequence Number": {
1195         "type" : "number",
1196         "description": "advertise a sequence number for the actual Notification",
1197         "x-ocf-conversion": {
1198             "x-ocf-alias": "oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide,
1199 oic.r.sensor.smoke, oic.r.sensor.water",
1200             "x-to-ocf": [
1201                 "ocf.sequencenumber = Sequence Number"
1202             ],
1203             "x-from-ocf": [
1204                 "N/A"
1205             ]
1206         }
1207     }
1208 }
1209 }
1210 },
1211 "type": "object",
1212 "allof": [
1213     {"$ref": "#/definitions/zwave.operation.notificationcommandclass"}
1214 ],
1215 "required": ["V1 Alarm Type", "V1 Alarm Level", "Notification Status", "Notification Type",
1216 "Notification Event:State", "Sequence", "Event:State Parameters Length"]
1217 }

```

1218 9.10 User Code Command Class

1219 9.10.1 Derived model

1220 The derived model: "zwave.operation.usercodecommandclass".

1221 **9.10.2 Property definition**

1222 Table 28 provides the detailed per Property mapping for
 1223 "zwave.operation.usercodecommandclass".

1224 **Table 28 – The Property mapping for "zwave.operation.usercodecommandclass".**

Z-Wave Property name	OCF Resource	To OCF	From OCF
User Identifier	oic.r.lock.code	Used as an index in the lock code array. It is defined in ZWave as 0..255 (8 bit field).	useridentifier = oic.r.lock.code.lockCodeList[arrayIndex]
User ID Status	oic.r.lock.code	N/A	User ID Status = 0x01
lockCodeList	oic.r.lock.code	User Identifier = ZWave Command Class User Identifier oic.r.lock.code.lockCodeList[User Identifier] = User Code	User Identifier = locally persisted ZWave Command Class User Identifier associated with this Resource User Code = oic.r.lock.code.lockCodeList[User Identifier]

1225 Table 29 provides the details of the Properties that are part of
 1226 "zwave.operation.usercodecommandclass".

1227 **Table 29 – The Properties of "zwave.operation.usercodecommandclass".**

Z-Wave Property name	Type	Required	Description
User Identifier	Number	yes	specify the actual User Identifier
User ID Status	Integer	yes	indicates the status of the User Identifier
lockCodeList	array	no	advertise the User Code to be set for the User Identifier

1228 **9.10.3 Derived model definition**

```

1229 {
1230   "id":
1231   "http://openinterconnect.org/zwavemapping/schemas/zwave.operation.usercodecommandclass.json#",
1232   "$schema": "http://json-schema.org/draft-04/schema#",
1233   "description": "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
1234   "title": "User Code Command Class",
1235   "definitions": {
1236     "zwave.operation.usercodecommandclass": {
1237       "type": "object",
1238       "properties": {
1239         "User Identifier": {
1240           "type": "Number",
1241           "description": "specify the actual User Identifier",
1242           "x-ocf-conversion": {
1243             "x-ocf-alias": "oic.r.lock.code",
1244             "x-to-ocf": [
1245               "Used as an index in the lock code array. It is defined in ZWave as 0..255 (8 bit
1246               field).",
1247             ],
1248             "x-from-ocf": [
1249               "useridentifier = oic.r.lock.code.lockCodeList[arrayIndex]"
1250             ]
1251           }
1252         },
1253         "User ID Status": {
1254           "type": "Integer",

```

```

1255         "description": "indicates the status of the User Identifier",
1256         "x-ocf-conversion": {
1257             "x-ocf-alias": "oic.r.lock.code",
1258             "x-to-ocf": [
1259                 "N/A "
1260             ],
1261             "x-from-ocf": [
1262                 "User ID Status = 0x01"
1263             ]
1264         }
1265     },
1266     "lockCodeList": {
1267         "type" : "array",
1268         "description": "advertise the User Code to be set for the User Identifier",
1269         "x-ocf-conversion": {
1270             "x-ocf-alias": "oic.r.lock.code",
1271             "x-to-ocf": [
1272                 "User Identifier = ZWave Command Class User Identifier",
1273                 "oic.r.lock.code.lockCodeList[User Identifier] = User Code"
1274             ],
1275             "x-from-ocf": [
1276                 "User Identifier = locally persisted ZWave Command Class User Identifier associated
1277 with this Resource",
1278                 "User Code = oic.r.lock.code.lockCodeList[User Identifier]"
1279             ]
1280         }
1281     }
1282 }
1283 }
1284 },
1285 "type": "object",
1286 "allOf": [
1287     {"$ref": "#/definitions/zwave.operation.doorlockoperationcommandclass"}
1288 ],
1289 "required": ["User Identifier", "User ID Status", "User Code"]
1290 }

```