OCF 2.3 – Z-Wave Data Model Mapping specification – BTG

Legal Disclaimer

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

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

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

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

CONTENTS

Figures

No table of figures entries found.

Tables

OPEN CONNECTIVITY FOUNDATION | Copyright Open Connectivity Foundation © 2017, 2018. All Rights Reserved.

1 Scope

The OCF Z-Wave Data Model specification ("this specification") provides detailed mapping information to provide equivalency between Z-Wave Command Classes and OCF defined Resources.

This specification provides mapping for Device Types (Z-Wave to OCF), identifies OCF Resources for both Z-Wave Command Classes and for each Command Class defines the detailed Property by Property mapping using OCF defined extensions to JSON schema to programmatically define the mappings.

2 Normative references

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

JSON Schema Core, *JSON Schema: core definitions and terminology*, January 2013
http://json-schema.org/latest/json-schema-core.html

JSON Schema Validation, *JSON Schema: interactive and non-interactive validation*, January 2013
http://json-schema.org/latest/json-schema-validation.html

JSON Hyper-Schema, *JSON Hyper-Schema: A Vocabulary for Hypermedia Annotation of JSON*, October 2016
http://json-schema.org/latest/json-schema-hypermedia.html

OCF Core Specification, *Open Connectivity Foundation Core Specification*, Version 2.0
https://openconnectivity.org/specs/OCF_Core_Specification_v2.0.0.pdf

OCF Device Specification, *Open Connectivity Foundation Device Specification, Version 2.0*
https://openconnectivity.org/specs/OCF_Device_Specification_v2.0.0.pdf

OCF Resource Type Specification, *Open Connectivity Foundation Resource Type Specification, Version 2.0*
https://openconnectivity.org/specs/OCF_Resource_Type_Specification_v2.0.0.pdf

OCF Security Specification, *Open Connectivity Foundation Security Specification*, Version 2.0
https://openconnectivity.org/specs/OCF_Security_Specification_v2.0.0.pdf

OCF Bridging Specification, *Open Connectivity Foundation Bridging Specification*, Version 2.0
https://openconnectivity.org/specs/OCF_Bridging_Specification_v1.3.0.pdf

RAML Specification, *RESTful API Modeling Language*, Version 0.8
https://github.com/raml-org/raml-spec/blob/master/versions/raml-08/raml-08.md

OpenAPI Specification, Version 2.0
https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md

Derived Models for Interoperability between IoT Ecosystems, Stevens & Merriam, March 2016
https://www.iab.org/wp-content/IAB-uploads/2016/03/OCF-Derived-Models-for-Interoperability-Between-IoT-Ecosystems_v2-examples.pdf

Z-Wave Plus Device and Command Class Types and Defines Specification

https://www.silabs.com/documents/login/miscellaneous/SDS11847-Z-Wave-Plus-Device-Type-Specification.pdf

Z-Wave Plus v2 Device Type Specification

https://www.silabs.com/documents/login/miscellaneous/SDS14224-Z-Wave-Plus-v2-Device-Type-Specification.pdf

3 Terms, definitions and symbols

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

3.1 Terms and definitions

As defined in the OCF Core Specification and OCF Bridging Specification with the following additions.

3.2 Terms and definitions

None defined.

3.3 Symbols and abbreviations

None defined.

4 Document conventions and organization

4.1 Introduction

For the purposes of this document, the terms and definitions given in OCF Core Specification and OCF Security Specification apply.

4.2 Conventions

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

4.3 Notation

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

Required (or shall or mandatory).

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

Recommended (or should).

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

Allowed (or allowed).

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

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

Conditionally required (CR)

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

DEPRECATED

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

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

Words that are emphasized are printed in *italic*.

4.4 Data types

Data types are defined in the OCF Core Specification.

5 Theory of Operation

5.1 Interworking Approach

The interworking between Z-Wave defined Command Classes and OCF defined Resource Types is modelled using the derived model syntax described in Derived Models for Interoperability.

5.2 Mapping Syntax

Within the defined syntax for derived modelling used by this Specification there are two blocks that define the actual Property-Property equivalence or mapping. These blocks are identified by the keywords 'x-to-ocf' and 'x-from-ocf'. Derived Models for Interoperability does not define a rigid syntax for these blocks; they are free form string arrays that contain pseudo-coded mapping logic. Within this specification we apply the rules in the following sub-sections to these blocks to ensure consistency and re-usability and extensibility of the mapping logic that is defined.

5.2.1 General

All statements are terminated with a carriage return.

5.2.2 Value Assignment

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

5.2.3 Property Naming

All Property names are identical to the name used by the original model; for example, from the OCF Temperature Resource the Property name 'temperature' is used whereas when referred to the derived ecosystem then the semantically equivalent Property name is used.
When the same name is used by both OCF and the derived ecosystem for semantically equivalent values then the name of the OCF defined Property is prepended by the ecosystem designator 'ocf' to avoid ambiguity (e.g. 'ocf.step')

5.2.4 Arrays

An array element is indicated by the use of square brackets '[]' with the index of the element contained therein, e.g. range[1]. All arrays start at an index of 0. If an entire array is being referenced then no index is included, e.g. selectablehumiditylevels[].

5.2.5 Default Mapping

There are cases where the specified mapping is not possible as one or more of the Properties being mapped is optional in the source model. In all such instances a default mapping is provided. The default map is indicated by the prepending of an 'otherwise:' modifier to the assignment. (e.g. 'otherwise: step = 1')

5.2.6 Conditional Mapping

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

if 'condition', 'mapping'.

Is applied.

E.g. if step >0, ocf.step = step.

5.2.7 Loops

When a mapping can be represented by a repeated loop governed by some condition then the syntax:

for 'initialize', 'condition', 'increment': 'mapping'

Where:

'initialize' is an initial local loop control variable setting.

'condition' is the loop controller, the loop repeats until the condition evaluates to 'false'.

'increment' allows for update of the control variable, if omitted an increment of '1' is assumed.

Is applied.

E.g. for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] = modearray[supportedmodes[x]]

5.2.8 Command Invocation

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

6 Device Type Mapping

6.1 Introduction

This Section contains the mappings to Device Types.

6.2 Z-Wave Device Types to OCF Device Types

The following table captures the mapping between Z-Wave Plus defined Device Types (see Z-Wave Plus Device Type Specification) and OCF defined Device Types (see OCF Device Specification).

Table 1. Z-Wave to OCF Device Type Mapping

Classification of Z-Wave Generic Type	Z-Wave Device Type	ZWave 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

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

7 Resource to Command Class Mapping

7.1 Introduction

Clause 7 lists the set of applicable Z-Wave Command Classes and provides the OCF Resource Type(s) to which the Command Classes map along an introduction the semantics of the mapping. The detailed mappings are provided in clause 8.

7.2 Z-Wave Command Classes to OCF Resources

The following tables capture the mapping between Z-Wave Command Classes and OCF defined Resource Types. Detailed Property by Property mappings are provided in Section 8.

Table 2 captures the mappings for Command Classes for a Z-Wave Device.

Table 2. 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	oic.wk.d		
	Version Command Class Z-Wave Plus Info Command Class	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	oic.wk.d		
	Version Command Class Z-Wave Plus Info Command Class	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 Z-Wave Plus Info Command Class	oic.wk.d oic.wk.p		
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		

7.2.1 Battery Command Class Mapping

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

Battery Energy to ‘charge’ is handled via the 'Battery Level' of Battery Command Class. A RETRIEVE on a Batter Energy maps to Battery Get Command on an instance of a Z-Wave Battery Command Class.

7.2.2 Binary Switch Command Class Mapping

This API defines the mapping between an instance of a Z-Wave Binary Switch Command Class and an OCF Binary Switch Resource. Note that the setting of the Value of OCF Binary Switch to '0x00' (off) and ‘0x255(on) is handled via the 'Value' of Binary Switch Command Class. A RETRIEVE on a Binary Switch maps to Binary Switch Get Command on an instance of a Z-Wave

Binary Switch Command Class. And a UPDATE on a Binary Switch maps to Binary Switch Set Command on an instance of a Z-Wave Binary Switch Command Class.

7.2.3 Door Lock Command Class Mapping

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

7.2.4 Multilevel Sensor Command Class Mapping

7.2.4.1 Multilevel Sensor Command Class Mapping for Carbon Dioxide Sensor

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

7.2.4.2 Multilevel Sensor Command Class Mapping for Carbon Monoxide Sensor

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

7.2.4.3 Multilevel Sensor Command Class Mapping for Smoke Density Sensor

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

7.2.4.4 Multilevel Sensor Command Class Mapping for Water Flow Sensor

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

7.2.5 Multilevel Switch Command Class Mapping

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

7.2.6 Notification Command Class Mapping

This API defines the mapping between an instance of a Notification Command Class and OCF Specific sensor resources. Notification Command Class has 9 properties; these map as follows: V1 Alarm Type, V1 Alarm Level, Notification Status, Notification Type, Notification Event:State,

Sequence, Event:State Parameters Length, Event:State Parameter, Sequence Number => corresponding properties of smoke sensor, carbon monoxide sensor, carbon dioxide sensor or water sensor. This is presented in OCF as the distinct Resource instances. A RETRIEVE on a Specific Sensor maps to Notification Get Command on an instance of a Z-Wave Notification Command Class. And a UPDATE on a Specific Sensor maps to Notification Set Command on an instance of a Z-Wave Notification Command Class.

7.2.7 User Code Command Class Mapping

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

8 Detailed Mapping APIs

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

Table 3 provides a reference and link to the per Command Class sub-sections.

Table 3. Command Class to Resource Summary

Z-Wave Command Class Name	Mapped Resource(s)	Mapping Section
Battery Command Class	oic.r.energy.battery	8.1
Binary Switch Command Class	oic.r.switch.binary	8.2
Door Lock Command Class	oic.r.lock.status	8.3
Multilevel Sensor Command Class	oic.r.sensor.carbondioxide	8.4
Multilevel Sensor Command Class	oic.r.sensor.carbonmonoxide	8.5
Multilevel Sensor Command Class	oic.r.sensor.water	8.6
Multilevel Sensor Command Class	oic.r.sensor.smoke	8.7
Multilevel Switch Command Class	oic.r.switch.binary	8.8
	oic.r.light.dimming	
Notification Command Class	oic.r.sensor.carbondioxide	8.9
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	8.10

8.1 Battery Command Class

8.1.1 Derived model

The derived model: zwave.operation.batterycommandclass

8.1.2 Property definition

Table 4 The property mapping for zwave.operation.batterycommandclass

ZWave 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

Table 5 The properties of zwave.operation.batterycommandclass

ZWave Property name	Type	Required	Description
Battery Level	if Battery Level = 255, stringif Battery Level != 255, integer	yes	percentage indicating the battery level or low battery warning

8.1.3 Derived model definition

```
{
  "id":
"http://openinterconnect.org/zwavemapping/schemas/zwave.operation.batterycommandclass.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Battery Command Class",
  "definitions": {
    "zwave.operation.batterycommandclass": {
      "type": "object",
      "properties": {
        "Battery Level": {
            "type": [
                "if Battery Level = 255, string",
                "if Battery Level != 255, integer"
                ],
          "description": "percentage indicating the battery level or low battery warning",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.energy.battery",
            "x-to-ocf": [
              "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."
            ],
            "x-from-ocf": [
                          "N/A"
                      ]
          }
        }
      }
```

```
    }
  },
  "type": "object",
  "allOf": [
    {"$ref": "#/definitions/zwave.operation.batterycommandclass"}
  ],
  "required": ["Battery Level"]
}
```

8.2 Binary Switch Command Class

8.2.1 Derived model

The derived model: zwave.operation.binaryswitchcommandclass

8.2.2 Property definition

Table 6 The property mapping for zwave.operation.binaryswitchcommandclass

ZWave 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

Table 7 The properties of zwave.operation.binaryswitchcommandclass

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

8.2.3 Derived model definition

```
{
  "id":
"http://openinterconnect.org/zwavemapping/schemas/zwave.operation.binaryswitchcommandclass.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Binary Switch Command Classs",
  "definitions": {
    "zwave.operation.binaryswitchcommandclass": {
      "type": "object",
      "properties": {
        "Value": {
          "type" :  "boolean",
          "description": "On/Off state at the receiving node",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.switch.binary",
            "x-to-ocf": [
              "if Value = 255, ocf.r.switch.binary.value = true.",
              "if Value != 255, ocf.r.switch.binary.value = false."
            ],
            "x-from-ocf": [
              "if ocf.r.switch.binary.value = false, Value = 0",
              "if ocf.r.switch.binary.value = true, Value = 255"

                        ]
          }
        }
      }
    }
  },
  "type": "object",
   "allOf": [
    {"$ref": "#/definitions/zwave.operation.binaryswitchcommandclass"}
  ],
```

```
  "required": ["Value"]
}
```

8.3 Door Lock Command Class

8.3.1 Derived model

The derived model: zwave.operation.doorlockcommandclass

8.3.2 Property definition

Table 8 The property mapping for zwave.operation.doorlockcommandclass

ZWave 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

Table 9 The properties of zwave.operation.doorlockcommandclass

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

8.3.3 Derived model definition

```
{
  "id":
"http://openinterconnect.org/zwavemapping/schemas/zwave.operation.doorlockcommandclass.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Door Lock Command Class",
  "definitions": {
    "zwave.operation.doorlockcommandclass": {
      "type": "object",
      "properties": {
        "Door Lock Mode": {
          "type" :  "integer",
          "description": "operation mode of the door lock device",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.lock.status",
            "x-to-ocf": [
              "if Door Lock Mode = 0x00, ocf.r.lock.status.lockState = UnLocked",
              "if Door Lock Mode = 0xFF, ocf.r.lock.status.lockState = Locked"
            ],
            "x-from-ocf": [
              "if ocf.r.lock.status.lockState = Unlocked, Door Lock Mode = 0x00",
              "if ocf.r.lock.status.lockState = Locked, Door Lock Mode = 0xFF"
                      ]
          }
        }
      }
    }
  },
  "type": "object",
   "allOf": [
    {"$ref": "#/definitions/zwave.operation.doorlockcommandclass"}
  ],
  "required": ["Door Lock Mode"]
}
```

8.4 Multilevel Sensor Command Class Carbon Dioxide

8.4.1 Derived model

The derived model: zwave.operation.multilevelsensorcommandclasscarbondioxide

8.4.2 Property definition

Table 10 The property mapping for zwave.operation.multilevelsensorcommandclasscarbondioxide

ZWave Property name	OCF Resource	To OCF	From OCF
Precision	oic.r.sensor.carbondioxide	ocf.r.sensor.carbondioxide.precision = Precision	N/A
Sensor Type	oic.r.sensor.carbondioxide	if Sensor Type = Carbon dioxide CO2-level, ocf.rt = oic.r.sensor.carbondioxide.	N/A
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
Scale	oic.r.sensor.carbondioxide	N/A	Scale = ppm (0x00)

Table 11 The properties of zwave.operation.multilevelsensorcommandclasscarbondioxide

ZWave Property name	Type	Required	Description
Precision	Number	yes	indicate how many decimal places are included the Sensor Value field
Sensor Type	Integer	yes	specify the carbon dioxide sensor type of 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
Scale	Integer	yes	indicate what scale is used for the actual sensor reading

8.4.3 Derived model definition

```
{
  "id":
"http://openinterconnect.org/zwavemapping/schemas/zwave.operation.multilevelsensorcommandclasscarbon
dioxide.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Multilevel Sensor Command Class Carbon Dioxide",
  "definitions": {
    "zwave.operation.multilevelsensorcommandclasscarbondioxide": {
      "type": "object",
      "properties": {
        "Sensor Type": {
```

```
            "type" :  "Integer",
            "description": " specify the carbon dioxide sensor type of the actual sensor reading ",
            "x-ocf-conversion": {
              "x-ocf-alias": "oic.r.sensor.carbondioxide",
              "x-to-ocf": [
                "if Sensor Type = Carbon dioxide CO2-level, ocf.rt = oic.r.sensor.carbondioxide."
              ],
              "x-from-ocf": [
                            "N/A"
                        ]
            }
          },
          "Precision": {
             "type" :  "Number",
             "description": " indicate how many decimal places are included the Sensor Value field ",
             "x-ocf-conversion": {
              "x-ocf-alias": "oic.r.sensor.carbondioxide",
               "x-to-ocf": [
                 "ocf.r.sensor.carbondioxide.precision = Precision"
               ],
              "x-from-ocf": [
                            "N/A"
                        ]
             }
           },
           "Scale": {
             "type" :  "Integer",
             "description": " indicate what scale is used for the actual sensor reading ",
             "x-ocf-conversion": {
              "x-ocf-alias": "oic.r.sensor.carbondioxide",
               "x-to-ocf": [
                 "N/A"
               ],
              "x-from-ocf": [
                            "Scale = ppm (0x00)"
                        ]
             }
           },
           "Size": {
             "type" :  "enum",
             "description": " indicate the length in bytes of the Sensor Value field ",
             "x-ocf-conversion": {
              "x-ocf-alias": "oic.r.sensor.carbondioxide",
               "x-to-ocf": [
                 "N/A"
               ],
              "x-from-ocf": [
                           "N/A"
                        ]
             }
           },
           "Sensor Value": {
             "type" :  "array",
             "description": " specify the value of the actual sensor reading ",
             "x-ocf-conversion": {
              "x-ocf-alias": "oic.r.sensor.carbondioxide",
               "x-to-ocf": [
                 "ocf.r.sensor.carbondioxide.value = true",
                 "ocf.r.sensor.carbondioxide.measurement = Sensor Value"
               ],
              "x-from-ocf": [
                            "N/A"
                        ]
             }
           }
         }
       }
     },
     "type": "object",
      "allOf": [
       {"$ref": "#/definitions/zwave.operation.multilevelsensorcommandclasscarbondioxide"}
```

```
    ],
    "required": ["Sensor Type", "Precision", "Scale", "Size", "Sensor Value"]
}
```

8.5 Multilevel Sensor Command Class Carbon Monoxide

8.5.1 Derived model

The derived model: zwave.operation.multilevelsensorcommandclasscarbonmonoxide

8.5.2 Property definition

Table 12 The property mapping for zwave.operation.multilevelsensorcommandclasscarbonmonoxide

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

Table 13 The properties of zwave.operation.multilevelsensorcommandclasscarbonmonoxide

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

8.5.3 Derived model definition

```
{
  "id":
"http://openinterconnect.org/zwavemapping/schemas/zwave.operation.multilevelsensorcommandclasscarbon
monoxide.json#",
```

```
628     "$schema": "http://json-schema.org/draft-04/schema#",
629     "description" : "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
630     "title": "Multilevel Sensor Command Class Carbon Monoxide",
631     "definitions": {
632       "zwave.operation.multilevelsensorcommandclasscarbonmonoxide": {
633         "type": "object",
634         "properties": {
635           "Sensor Type": {
636             "type" :  "Integer",
637             "description": " specify the carbon monoxidesensor type of the actual sensor reading ",
638             "x-ocf-conversion": {
639               "x-ocf-alias": "oic.r.sensor.carbonmonoxide",
640               "x-to-ocf": [
641                 "if Sensor Type = Carbon monoxide (CO) level, ocf.rt = oic.r.sensor.carbonmonoxide."
642               ],
643               "x-from-ocf": [
644                             "N/A"
645                         ]
646             }
647           },
648           "Precision": {
649              "type" :  "Number",
650              "description": " indicate how many decimal places are included the Sensor Value field ",
651              "x-ocf-conversion": {
652               "x-ocf-alias": "oic.r.sensor.carbonmonoxide",
653                "x-to-ocf": [
654                  "ocf.r.sensor.carbonmonoxide.precision = Precision"
655                ],
656               "x-from-ocf": [
657                             "N/A"
658                         ]
659             }
660           },
661           "Scale": {
662             "type" :  "Integer",
663             "description": " indicate what scale is used for the actual sensor reading ",
664             "x-ocf-conversion": {
665              "x-ocf-alias": "oic.r.sensor.carbonmonoxide",
666               "x-to-ocf": [
667                 "N/A"
668               ],
669              "x-from-ocf": [
670                             "Scale = ppm (0x01)"
671                         ]
672             }
673           },
674           "Size": {
675             "type" :  "enum",
676             "description": " indicate the length in bytes of the Sensor Value field ",
677             "x-ocf-conversion": {
678              "x-ocf-alias": "oic.r.sensor.carbonmonoxide",
679               "x-to-ocf": [
680                 "N/A"
681               ],
682              "x-from-ocf": [
683                             "N/A"
684                         ]
685             }
686           },
687           "Sensor Value": {
688             "type" :  "array",
689             "description": " specify the value of the actual sensor reading ",
690             "x-ocf-conversion": {
691              "x-ocf-alias": "oic.r.sensor.carbonmonoxide",
692               "x-to-ocf": [
693                 "ocf.r.sensor.carbonmonoxide.value = true",
694                 "ocf.r.sensor.carbonmonoxide.measurement = Sensor Value"
695               ],
696              "x-from-ocf": [
697                             "N/A"
698                         ]
```

```
            }
          }
        }
      }
    },
    "type": "object",
     "allOf": [
      {"$ref": "#/definitions/zwave.operation.multilevelsensorcommandclasscarbonmonoxide"}
    ],
    "required": ["Sensor Type", "Precision", "Scale", "Size", "Sensor Value"]
}
```

8.6 Multilevel Sensor Command Class Smoke Density

8.6.1 Derived model

The derived model: zwave.operation.multilevelsensorcommandclasssmokedensity

8.6.2 Property definition

Table 14 The property mapping for zwave.operation.multilevelsensorcommandclasssmokedensity

ZWave Property name	OCF Resource	To OCF	From OCF
Sensor Value	oic.r.sensor.smoke	ocf.r.sensor.smoke.value = trueocf.r.sensor.smoke.measurement = Sensor Value	N/A
Size	oic.r.sensor.smoke	N/A	N/A
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)

Table 15 The properties of zwave.operation.multilevelsensorcommandclasssmokedensity

ZWave Property name	Type	Required	Description
Sensor Value	array	yes	specify the value of the actual sensor reading
Size	enum	yes	indicate the length in bytes of the Sensor Value field
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

8.6.3 Derived model definition

```
{
  "id":
```

```
"http://openinterconnect.org/zwavemapping/schemas/zwave.operation.multilevelsensorcommandclasssmokedensity.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Multilevel Sensor Command Class Smoke Density",
  "definitions": {
    "zwave.operation.multilevelsensorcommandclasssmokedensity": {
      "type": "object",
      "properties": {
        "Sensor Type": {
          "type" :  "Integer",
          "description": " specify the smoke density sensor type of the actual sensor reading ",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.sensor.smoke",
            "x-to-ocf": [
              "if Sensor Type = Smoke density, ocf.rt = oic.r.sensor.smoke."
            ],
            "x-from-ocf": [
                          "N/A"
                    ]
          }
        },
        "Precision": {
           "type" :  "Number",
           "description": " indicate how many decimal places are included the Sensor Value field ",
           "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.sensor.smoke",
             "x-to-ocf": [
               "ocf.r.sensor.smoke.precision = Precision"
             ],
            "x-from-ocf": [
                          "N/A"
                    ]
          }
         },
         "Scale": {
           "type" :  "Integer",
           "description": " indicate what scale is used for the actual sensor reading ",
           "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.sensor.smoke",
             "x-to-ocf": [
               "N/A"
             ],
            "x-from-ocf": [
                          "Scale = percent (0x00)"
                    ]
          }
         },
         "Size": {
           "type" :  "enum",
           "description": " indicate the length in bytes of the Sensor Value field ",
           "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.sensor.smoke",
             "x-to-ocf": [
               "N/A"
             ],
            "x-from-ocf": [
                         "N/A"
                    ]
          }
         },
         "Sensor Value": {
           "type" :  "array",
           "description": " specify the value of the actual sensor reading ",
           "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.sensor.smoke",
             "x-to-ocf": [
               "ocf.r.sensor.smoke.value = true",
               "ocf.r.sensor.smoke.measurement = Sensor Value"
             ],
            "x-from-ocf": [
```

```
                            "N/A"
                        ]
                    }
                }
            }
        }
    },
    "type": "object",
     "allOf": [
      {"$ref": "#/definitions/zwave.operation.multilevelsensorcommandclasssmokedensity"}
    ],
    "required": ["Sensor Type", "Precision", "Scale", "Size", "Sensor Value"]
}
```

8.7 Multilevel Sensor Command Class Water Flow

8.7.1 Derived model

The derived model: zwave.operation.multilevelsensorcommandclasswaterflow

8.7.2 Property definition

Table 16 The property mapping for zwave.operation.multilevelsensorcommandclasswaterflow

ZWave 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
Scale	oic.r.sensor.water	N/A	Scale = litre/hr (0x00)
Sensor Value	oic.r.sensor.water	ocf.r.sensor.water.value = true ocf.r.sensor.water.measurement = Sensor Value	N/A
Precision	oic.r.sensor.water	ocf.r.sensor.water.precision = Precision	N/A
Size	oic.r.sensor.water	N/A	N/A

Table 17 The properties of zwave.operation.multilevelsensorcommandclasswaterflow

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

8.7.3 Derived model definition


```
{
  "id":
"http://openinterconnect.org/zwavemapping/schemas/zwave.operation.multilevelsensorcommandclasswaterf
low.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Multilevel Sensor Command Class Water Flow",
  "definitions": {
    "zwave.operation.multilevelsensorcommandclasswaterflow": {
      "type": "object",
      "properties": {
        "Sensor Type": {
          "type" :  "Integer",
          "description": " specify the water flow sensor type of the actual sensor reading ",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.sensor.water",
            "x-to-ocf": [
              "if Sensor Type = Water flow, ocf.rt = oic.r.sensor.water."
            ],
            "x-from-ocf": [
                          "N/A"
                    ]
          }
        },
        "Precision": {
           "type" :  "Number",
           "description": " indicate how many decimal places are included the Sensor Value field ",
           "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.sensor.water",
             "x-to-ocf": [
               "ocf.r.sensor.water.precision = Precision"
             ],
            "x-from-ocf": [
                          "N/A"
                    ]
          }
         },
         "Scale": {
           "type" :  "Integer",
           "description": " indicate what scale is used for the actual sensor reading ",
           "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.sensor.water",
             "x-to-ocf": [
               "N/A"
             ],
            "x-from-ocf": [
                          "Scale = litre/hr (0x00)"
                    ]
          }
         },
         "Size": {
           "type" :  "enum",
           "description": " indicate the length in bytes of the Sensor Value field ",
           "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.sensor.water",
             "x-to-ocf": [
               "N/A"
             ],
            "x-from-ocf": [
                          "N/A"
                    ]
          }
         },
         "Sensor Value": {
           "type" :  "array",
           "description": " specify the value of the actual sensor reading ",
           "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.sensor.water",
             "x-to-ocf": [
```

```
                "ocf.r.sensor.water.value = true",
                "ocf.r.sensor.water.measurement = Sensor Value"
              ],
             "x-from-ocf": [
                          "N/A"
                        ]
            }
          }
        }
      }
    },
    "type": "object",
     "allOf": [
      {"$ref": "#/definitions/zwave.operation.multilevelsensorcommandclasswaterflow"}
    ],
    "required": ["Sensor Type", "Precision", "Scale", "Size", "Sensor Value"]
}
```

8.8 Multilevel Switch Command Class

8.8.1 Derived model

The derived model: zwave.operation.multilevelswitchcommandclass

8.8.2 Property definition

Table 18 The property mapping for zwave.operation.multilevelswitchcommandclass

ZWave 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 = false otherwise: ocf.rt = oic.r.light.dimming; ocf.r.light.dimming.dimmingSetting = value	value = dimmingSetting if ocf.rt = oic.r.switch.binary, value = ocf.r.switch.binary.value if ocf.rt = oic.r.light.dimming, value = dimmingSetting

Table 19 The properties of zwave.operation.multilevelswitchcommandclass

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

8.8.3 Derived model definition

```
{
  "id":
"http://openinterconnect.org/zwavemapping/schemas/zwave.operation.multilevelswtichcommandclass.json#
",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Multilevel Switch Command Class",
  "definitions": {
    "zwave.operation.multilevelswitchcommandclass": {
      "type": "object",
      "properties": {
        "Value": {
          "type" : "integer, boolean",
          "description": "multilevel value in a supporting device",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.switch.binary, oic.r.light.dimming",
            "x-to-ocf": [
              "if value = 0, ocf.rt = oic.r.switch.binary & ocf.r.switch.binary.value = false",
```

```
                "otherwise: ocf.rt = oic.r.light.dimming; ocf.r.light.dimming.dimmingSetting = value"
              ],
              "x-from-ocf": [
                "value = dimmingSetting",
                "if ocf.rt = oic.r.switch.binary, value = ocf.r.switch.binary.value",
                "if ocf.rt = oic.r.light.dimming, value = dimmingSetting"
                        ]
            }
          }
        }
      }
    },
    "type": "object",
    "allOf": [
      {"$ref": "#/definitions/zwave.operation.multilevelswitchcommandclass"}
    ],
    "required": ["Value"]
}
```

8.9 Notification Command Class

8.9.1 Derived model

The derived model: zwave.operation.notificationcommandclass

8.9.2 Property definition

Table 20 The property mapping for zwave.operation.notificationcommandclass

ZWave Property name	OCF Resource	To OCF	From OCF
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
V1 Alarm Type	oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide, oic.r.sensor.smoke, oic.r.sensor.water	N/A	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
Sequence	oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide, oic.r.sensor.smoke, oic.r.sensor.water	ocf.sequence = Sequence	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	N/A

		Notification Type = Water Alarm, ocf.rt = oic.r.sensor.water.	
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
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

949 **Table 21 The properties of zwave.operation.notificationcommandclass**

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

950 ### 8.9.3 Derived model definition

```
951 {
952   "id":
953 "http://openinterconnect.org/zwavemapping/schemas/zwave.operation.notificationcommandclass.json#",
954   "$schema": "http://json-schema.org/draft-04/schema#",
955   "description" : "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
956   "title": "Notification Command Class",
957   "definitions": {
958     "zwave.operation.notificationcommandclass": {
959       "type": "object",
```

```
960         "properties": {
961           "V1 Alarm Type": {
962             "type" :  "Integer",
963             "description": "depends on the V1 Alarm field advertised in the Alarm Type Supported
964    Report Command",
965             "x-ocf-conversion": {
966              "x-ocf-alias": "oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide,
967    oic.r.sensor.smoke, oic.r.sensor.water",
968              "x-to-ocf": [
969                "N/A"
970              ],
971               "x-from-ocf": [
972                           "N/A"
973                         ]
974             }
975           },
976           "V1 Alarm Level": {
977             "type" :  "Integer",
978             "description": "product manual specific",
979             "x-ocf-conversion": {
980              "x-ocf-alias": "oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide,
981    oic.r.sensor.smoke, oic.r.sensor.water",
982               "x-to-ocf": [
983                  "N/A"
984               ],
985               "x-from-ocf": [
986                             "N/A"
987                         ]
988             }
989           },
990           "Notification Status": {
991              "type" :  "Integer",
992              "description": "advertise the status of the Notification Type",
993              "x-ocf-conversion": {
994              "x-ocf-alias": "oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide,
995    oic.r.sensor.smoke, oic.r.sensor.water",
996                "x-to-ocf": [
997                  "Value = Notification Status"
998                ],
999               "x-from-ocf": [
1000                            "N/A"
1001                        ]
1002             }
1003           },
1004           "Notification Type": {
1005             "type" :  "Integer",
1006             "description": " specify a Notification Type ",
1007             "x-ocf-conversion": {
1008             "x-ocf-alias": "oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide,
1009   oic.r.sensor.smoke, oic.r.sensor.water",
1010               "x-to-ocf": [
1011                 "if Notification Type = Smoke Alarm, ocf.rt = oic.r.sensor.smoke.",
1012                 "if Notification Type = CO Alarm, ocf.rt = oic.r.sensor.carbonmonoxide.",
1013                 "if Notification Type = CO2 Alarm, ocf.rt = oic.r.sensor.carbondioxide.",
1014                 "if Notification Type = Water Alarm, ocf.rt = oic.r.sensor.water."
1015               ],
1016               "x-from-ocf": [
1017                            "N/A"
1018                        ]
1019             }
1020           },
1021           "Notification Event:State": {
1022             "type" :  "Integer",
1023             "description": "specify a Notification Event/State for the advertised Notification Type",
1024             "x-ocf-conversion": {
1025             "x-ocf-alias": "oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide,
1026   oic.r.sensor.smoke, oic.r.sensor.water",
1027               "x-to-ocf": [
1028                 "Value = Notification Event:State"
1029               ],
1030              "x-from-ocf": [
```

```
                                "N/A"
                            ]
                        }
                    },
                    "Sequence": {
                        "type" :  "boolean",
                        "description": "advertise the presence of the Sequence Number field",
                        "x-ocf-conversion": {
                        "x-ocf-alias": "oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide,
oic.r.sensor.smoke, oic.r.sensor.water",
                            "x-to-ocf": [
                                "ocf.sequence = Sequence"
                            ],
                            "x-from-ocf": [
                                "N/A"
                            ]
                        }
                    },
                    "Event:State Parameters Length": {
                        "type" :  "number",
                        "description": "advertise the length in bytes of the Event / State Parameters field",
                        "x-ocf-conversion": {
                        "x-ocf-alias": "oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide,
oic.r.sensor.smoke, oic.r.sensor.water",
                            "x-to-ocf": [
                                "ocf.event:stateparameterslength = Event:State Parameters Length"
                            ],
                            "x-from-ocf": [
                                "N/A"
                            ]
                        }
                    },
                    "Event:State Parameter": {
                        "type" :  "Integer",
                        "description": "specify associated parameters to a Notification",
                        "x-ocf-conversion": {
                        "x-ocf-alias": "oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide,
oic.r.sensor.smoke, oic.r.sensor.water",
                            "x-to-ocf": [
                                "ocf.event:stateparameter = Event:State Parameter"
                            ],
                            "x-from-ocf": [
                                "N/A"
                            ]
                        }
                    },
                    "Sequence Number": {
                        "type" :  "number",
                        "description": "advertise a sequence number for the actual Notification",
                        "x-ocf-conversion": {
                        "x-ocf-alias": "oic.r.sensor.carbondioxide, oic.r.sensor.carbonmonoxide,
oic.r.sensor.smoke, oic.r.sensor.water",
                            "x-to-ocf": [
                                "ocf.sequencenumber = Sequence Number"
                            ],
                            "x-from-ocf": [
                                "N/A"
                            ]
                        }
                    }
                }
            }
        },
        "type": "object",
         "allOf": [
          {"$ref": "#/definitions/zwave.operation.notificationcommandclass"}
        ],
        "required": ["V1 Alarm Type", "V1 Alarm Level", "Notification Status", "Notification Type",
"Notification Event:State", "Sequence", "Event:State Parameters Length"]
    }
```

8.10 User Code Command Class

8.10.1 Derived model

The derived model: zwave.operation.usercodecommandclass

8.10.2 Property definition

Table 22 The property mapping for zwave.operation.usercodecommandclass

ZWave 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]

Table 23 The properties of zwave.operation.usercodecommandclass

ZWave 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

8.10.3 Derived model definition

```
{
  "id":
"http://openinterconnect.org/zwavemapping/schemas/zwave.operation.usercodecommandclass.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "User Code Command Class",
  "definitions": {
    "zwave.operation.usercodecommandclass": {
      "type": "object",
      "properties": {
        "User Identifier": {
          "type" :  "Number",
          "description": "specify the actual User Identifier",
          "x-ocf-conversion": {
             "x-ocf-alias": "oic.r.lock.code",
             "x-to-ocf": [
               "Used as an index in the lock code array. It is defined in ZWave as 0..255 (8 bit
field)."
             ],
            "x-from-ocf": [
               "useridentifier = oic.r.lock.code.lockCodeList[arrayIndex]"
                      ]
           }
         },
         "User ID Status": {
           "type" :  "Integer",
```

```
            "description": "indicates the status of the User Identifier",
            "x-ocf-conversion": {
              "x-ocf-alias": "oic.r.lock.code",
              "x-to-ocf": [
                "N/A "
              ],
             "x-from-ocf": [
                "User ID Status = 0x01"
                        ]
            }
          },
          "lockCodeList": {
            "type" :  "array",
            "description": "advertise the User Code to be set for the User Identifier",
            "x-ocf-conversion": {
              "x-ocf-alias": "oic.r.lock.code",
              "x-to-ocf": [
                "User Identifier = ZWave Command Class User Identifier",
                "oic.r.lock.code.lockCodeList[User Identifier] = User Code"
              ],
             "x-from-ocf": [
                "User Identifier = locally persisted ZWave Command Class User Identifier associated
with this Resource",
                "User Code = oic.r.lock.code.lockCodeList[User Identifier]"
                        ]
            }
          }
        }
      }
    },
    "type": "object",
     "allOf": [
      {"$ref": "#/definitions/zwave.operation.doorlockoperationcommandclass"}
    ],
    "required": ["User Identifier", "User ID Status", "User Code"]
}
```