OCF Resource to Z-Wave Mapping

VERSION 2.2.7 | November 2023

CONTACT admin@openconnectivity.org

Legal Disclaimer

CONTENTS

Figures

Tables

Introduction

This document, and all the other parts associated with this document, were developed in response to worldwide demand for smart home focused Internet of Things (IoT) devices, such as appliances, door locks, security cameras, sensors, and actuators; these to be modelled and securely controlled, locally and remotely, over an IP network.

While some inter-device communication existed, no universal language had been developed for the IoT. Device makers instead had to choose between disparate frameworks, limiting their market share, or developing across multiple ecosystems, increasing their costs. The burden then falls on end users to determine whether the products they want are compatible with the ecosystem they bought into, or find ways to integrate their devices into their network, and try to solve interoperability issues on their own.

In addition to the smart home, IoT deployments in commercial environments are hampered by a lack of security. This issue can be avoided by having a secure IoT communication framework, which this standard solves.

The goal of these documents is then to connect the next 25 billion devices for the IoT, providing secure and reliable device discovery and connectivity across multiple OSs and platforms. There are multiple proposals and forums driving different approaches, but no single solution addresses the majority of key requirements. This document and the associated parts enable industry consolidation around a common, secure, interoperable approach.

The OCF specification suite is made up of nineteen discrete documents, the documents fall into logical groupings as described herein:

- Core framework
  - Core Specification
  - Security Specification
  - Onboarding Tool Specification
- Bridging framework and bridges
  - Bridging Specification
  - Resource to Alljoyn Interface Mapping Specification
  - OCF Resource to oneM2M Resource Mapping Specification
  - OCF Resource to BLE Mapping Specification
  - OCF Resource to EnOcean Mapping Specification
  - OCF Resource to LWM2M Mapping Specification
  - OCF Resource to UPlus Mapping Specification
  - OCF Resource to Zigbee Cluster Mapping Specification
  - OCF Resource to Z-Wave Mapping Specification
- Resource and Device models
  - Resource Type Specification
  - Device Specification
- Core framework extensions
  - Easy Setup Specification
  - Core Optional Specification
- OCF Cloud
  - Cloud API for Cloud Services Specification

- Device to Cloud Services Specification
- Cloud Security Specification

OCF Resource to Z-Wave Mapping Specification

1 Scope

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

2 Normative references

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

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

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

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

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 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, symbols and abbreviated terms

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

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

– ISO Online browsing platform: available at https://www.iso.org/obp
– IEC Electropedia: available at http://www.electropedia.org/

3.1 Terms and definitions

3.1.1
Command Class
collection of commands used for controlling, querying, and reporting information corresponding to specific function supported by a Z-Wave device.

4 Document conventions and organization

4.1 Conventions

In this document 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.

In this document, to be consistent with the IETF usages for RESTful operations, the RESTful operation words CRUDN, CREATE, RETRIVE, UPDATE, DELETE, and NOTIFY will have all letters capitalized. Any lowercase uses of these words have the normal technical English meaning.

4.2 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 the Mapping Specification. The phrases "shall not", and "PROHIBITED" indicate behavior that is prohibited, i.e. that if performed means the implementation is not in compliance.

Recommended (or should).

> These features add functionality supported by the Mapping Specification and should be implemented. Recommended features take advantage of the capabilities the Mapping Specification, 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 behavior that is permitted but not recommended.

Allowed (or allowed).

> These features are neither required nor recommended by the Mapping Specification, 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 document, they should not be implemented except for backward compatibility. The occurrence of a deprecated feature during operation of

an implementation compliant with the current document 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 document.

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

Words that are emphasized are printed in *italic*.

5 Theory of operation

5.1 Interworking approach

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

5.2 Mapping syntax

5.2.1 Introduction

Within the defined syntax for derived modelling used by this document 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 between IoT Ecosystems does not define a rigid syntax for these blocks; they are free form string arrays that contain pseudo-coded mapping logic.

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

5.2.2 General

All statements are terminated with a carriage return.

5.2.3 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.

5.2.4 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.

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

5.2.5 Range

The range on the OCF side is fixed.

5.2.6 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.

5.2.7 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. (e.g. "transitiontime = 1")

5.2.8 Conditional mapping

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

If "condition", then "mapping".

is applied.

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

5.2.9 Method 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 resource, service, interface 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 Z-Wave translation

6.1 Operational scenarios

6.1.1 Introduction

The overall goals are to:

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

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

6.1.2 Overview of OCF-Z-Wave bridging

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

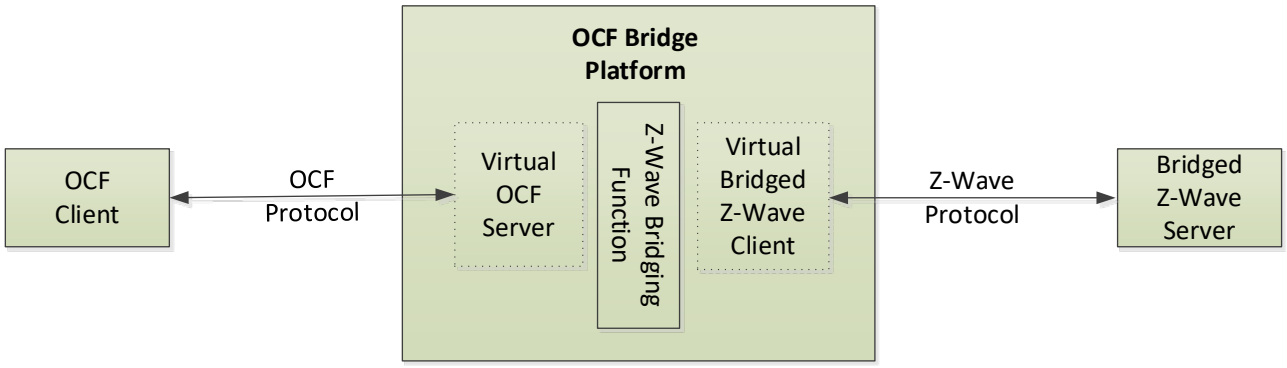

Figure 1 – OCF Z-Wave Bridge Platform and Components

6.1.3 Use case for OCF Client and Z-Wave server

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

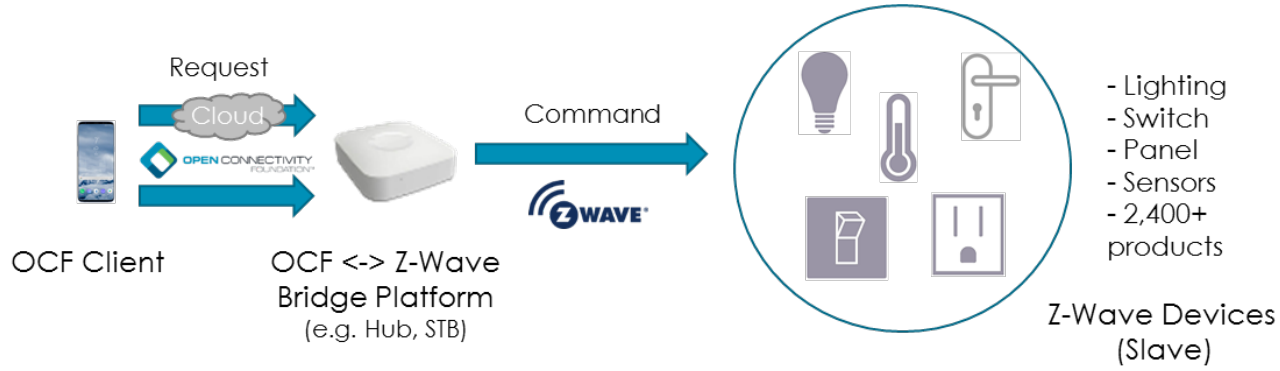

Figure 2 – OCF Client and Z-Wave Server

6.2 Requirements specific to Z-Wave bridging function

6.2.1 Requirements specific to Z-Wave

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

6.2.2 Exposing Z-Wave servers to OCF clients

6.2.2.1 General

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

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

Table 2is 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) Version Command Class (Version Get/Report) Z-Wave Plus Info Command Class (Z-Wave Plus Info Get/Report)		"oic.wk.d" (Device) "oic.wk.p" (Platform)
Z-Wave Command Parameter	Value (255 or 0)	**OCF Resource Property**	"value" (True or False)
	Value (1~99)		"dimmingSetting" (Integer)

If Z-Wave Plus device, Z-Wave Command Class, Z-Wave Command Parameter are enlisted in the well-defined set as specified in OCF Z-Wave Data Model Mapping, Bridging Function shall follow

the requirements for translating it to an OCF device, OCF resource or OCF resource property (i.e., “deep translation”).

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

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

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

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

6.2.2.2 Translation for well-defined set

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

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	oic.wk.d		
		oic.wk.p		

	Version Command Class Z-Wave Plus Info Command Class			
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		

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

6.2.2.3 Exposing a Z-Wave server as a virtual OCF server

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

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 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 30118-2for Device ID.	Y	(none)	Use as defined in ISO/IEC 30118-2	
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	N

					assigns a version number Defined in Version Command Class	
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

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

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	N	(none)		

		current region in which the device is located geographically. The free form text shall not start with a quote (").				
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

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

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 UUID in accordance with IETF RFC 4122. It is recommended that the UUID be created using the random generation scheme (version 4 UUID) specific in the RFC.	Y	(none)	Bridging Function should return a random-generated UUID as specified in the section 4.4 of IETF RFC 4122.	
Manufacturer Name	mnmn	Name of manufacturer (not to	Y	Translate Manufacturer ID to Human friendly name based upon the Manufacturer	Manufacturer ID: the unique ID identifying the manufacturer of the	Y

		exceed 16 characters)		ID/Manufacturer name table within Z-Wave Controller	device. Defined in Manufacturer Specific Command Class	
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	mnsl	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)		

6.2.2.4 On-the-fly translation

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

7 Device type mapping

7.1 Introduction

This clause contains the mappings to Device Types.

7.2 Z-Wave device types to OCF device types

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

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

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

8 Resource to command class mapping

8.1 Introduction

This clause 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 9.

8.2 Z-Wave command classes to OCF resources

8.2.1 Introduction

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

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

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 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		

8.2.2 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.

8.2.3 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.

8.2.4 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.

8.2.5 Multilevel sensor command class mapping

8.2.5.1 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.

8.2.5.2 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.

8.2.5.3 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.

8.2.5.4 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.

8.2.6 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.

8.2.7 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.

8.2.8 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.

9 Detailed mapping APIs

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

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

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

9.1 Battery command class

9.1.1 Derived model

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

9.1.2 Property definition

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

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

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

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

Z-Wave 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

9.1.3 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"]
}
```

9.2 Binary switch command class

9.2.1 Derived model

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

9.2.2 Property definition

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

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 = 0if ocf.r.switch.binary.value = true, Value = 255

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

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

9.2.3 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"]
}
```

9.3 Door lock command class

9.3.1 Derived model

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

9.3.2 Property definition

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

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 = UnLockedif Door Lock Mode = 0xFF, ocf.r.lock.status.lockState = Locked	if ocf.r.lock.status.lockState = Unlocked, Door Lock Mode = 0x00if ocf.r.lock.status.lockState = Locked, Door Lock Mode = 0xFF

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

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

9.3.3 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"]
}
```

9.4 Multilevel sensor command class carbon dioxide

9.4.1 Derived model

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

9.4.2 Property definition

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

Table 16 – The Property mapping for "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

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

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

9.4.3 Derived model definition

```
{
  "id":
"http://openinterconnect.org/zwavemapping/schemas/zwave.operation.multilevelsensorcommandclasscarbo
ndioxide.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"]
}
```

9.5 Multilevel sensor command class carbon monoxide

9.5.1 Derived model

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

9.5.2 Property definition

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

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

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

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

9.5.3 Derived model definition

```
{
  "id":
"http://openinterconnect.org/zwavemapping/schemas/zwave.operation.multilevelsensorcommandclasscarbo
nmonoxide.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 Monoxide",
  "definitions": {
    "zwave.operation.multilevelsensorcommandclasscarbonmonoxide": {
      "type": "object",
      "properties": {
        "Sensor Type": {
          "type" :  "Integer",
          "description": " specify the carbon monoxidesensor type of the actual sensor reading ",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.sensor.carbonmonoxide",
            "x-to-ocf": [
              "if Sensor Type = Carbon monoxide (CO) level, ocf.rt = oic.r.sensor.carbonmonoxide."
            ],
            "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.carbonmonoxide",
            "x-to-ocf": [
              "ocf.r.sensor.carbonmonoxide.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.carbonmonoxide",
            "x-to-ocf": [
              "N/A"
            ],
           "x-from-ocf": [
                          "Scale = ppm (0x01)"
                    ]
        }
       },
       "Size": {
          "type" :  "enum",
          "description": " indicate the length in bytes of the Sensor Value field ",
          "x-ocf-conversion": {
           "x-ocf-alias": "oic.r.sensor.carbonmonoxide",
            "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.carbonmonoxide",
            "x-to-ocf": [
```

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

9.6 Multilevel sensor command class smoke density

9.6.1 Derived model

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

9.6.2 Property definition

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

Table 20 – The Property mapping for "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

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

Table 21 – The Properties of "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

9.6.3 Derived model definition

```
{
  "id":
"http://openinterconnect.org/zwavemapping/schemas/zwave.operation.multilevelsensorcommandclasssmoke
density.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"]
}
```

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".

Table 22 – The Property mapping for "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

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

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

```
{
  "id":
"http://openinterconnect.org/zwavemapping/schemas/zwave.operation.multilevelsensorcommandclasswater
flow.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"]
}
```

9.8 Multilevel switch command class

9.8.1 Derived model

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

9.8.2 Property definition

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

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

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

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

9.8.3 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"]
}
```

9.9 Notification command class

9.9.1 Derived model

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

9.9.2 Property definition

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

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,	N/A	N/A

	oic.r.sensor.smoke, oic.r.sensor.water		
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

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

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

9.9.3 Derived model definition

```
{
  "id":
"http://openinterconnect.org/zwavemapping/schemas/zwave.operation.notificationcommandclass.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2018 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Notification Command Class",
  "definitions": {
    "zwave.operation.notificationcommandclass": {
      "type": "object",
      "properties": {
        "V1 Alarm Type": {
          "type" :  "Integer",
          "description": "depends on the V1 Alarm field advertised in the Alarm Type Supported
Report Command",
          "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": [
             "N/A"
           ],
            "x-from-ocf": [
                        "N/A"
                      ]
          }
        },
        "V1 Alarm Level": {
          "type" :  "Integer",
          "description": "product manual specific",
          "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": [
               "N/A"
            ],
            "x-from-ocf": [
                          "N/A"
                      ]
          }
        },
        "Notification Status": {
           "type" :  "Integer",
           "description": "advertise the status of the Notification Type",
           "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": [
               "Value = Notification Status"
             ],
            "x-from-ocf": [
                           "N/A"
                      ]
          }
         },
         "Notification Type": {
```

```
          "type" :  "Integer",
          "description": " specify a Notification Type ",
          "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": [
              "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."
            ],
            "x-from-ocf": [
                          "N/A"
                    ]
          }
        },
        "Notification Event:State": {
          "type" :  "Integer",
          "description": "specify a Notification Event/State for the advertised Notification
Type",
          "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": [
              "Value = Notification Event:State"
            ],
           "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"]
}
```

9.10 User code command class

9.10.1 Derived model

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

9.10.2 Property definition

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

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 Identifieroic.r.lock.code.lockCodeList[User Identifier] = User Code	User Identifier = locally persisted ZWave Command Class User Identifier associated with this ResourceUser Code = oic.r.lock.code.lockCodeList[User Identifier]

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

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

9.10.3 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"]
}
```