INTERNATIONAL STANDARD

ISO/IEC 30118-6

First edition
2018-11

Information technology — Open Connectivity Foundation (OCF) Specification —

Part 6: Resource to AllJoyn interface mapping specification

Technologies de l'information — Spécification de la Fondation pour la connectivité ouverte (Fondation OCF) —

Partie 6: Spécification du mapping entre les ressources et l'interface AllJoyn

Reference number
ISO/IEC 30118-6:2018(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by the Open Connectivity Foundation (OCF) (as the OCF Resource to AllJoyn Interface Mapping, Version 1.0.0) and drafted in accordance with its editorial rules. It was adopted, under the JTC 1 PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

A list of all parts in the ISO/IEC 30118 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

CONTENTS

Figures

No table of figures entries found.

Tables

1 Scope

The OCF Resource to AllJoyn Interface Mapping specification ("this specification") provides detailed mapping information to provide equivalency between AllJoyn defined Interfaces and OCF defined Resources,

This specification provides mapping for Device Types (AllJoyn to/from OCF), identifies equivalent OCF Resources for both mandatory and optional AllJoyn interfaces and for each interface 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.

OCF Core Specification, *Open Interconnect Consortium Core Specification*, Version 1.0.

OCF Resource Type Specification, *Open Interconnect Consortium Resource Type Specification*, Version 1.0

OCF Smart Home Device Specification, *Open Interconnect Consortium Smart Home Device Specification*, Version 1.0

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

IETF RFC 7159, *The JavaScript Object Notation (JSON) Data Interchange Format*, March 2014
http://www.ietf.org/rfc/rfc7159.txt

RAML, *Restful API modelling language,* Version 0.8.
https://github.com/raml-org/raml-spec/blob/master/versions/raml-08/raml-08.md

AllJoyn Common Data Model Interface Definitions
https://wiki.alljoyn.org/cdm

Swagger2.0, *Swagger RESTful API Documentation Specification*, Version 2.0
http://swagger.io/specification/

OCF Resource Type Definitions, *API Definition Language for OCF Resource Type Definitions,* Release OCF-v1.0.0
https://github.com/openconnectivityfoundation/derivedmodels

3 Terms, definitions symbols and abbreviations

3.1 Terms and definitions

3.2 Symbols and abbreviations

3.2.1

OCF
Open Connectivity Foundation
The organization that created these specifications

3.2.2

RAML
RESTful API Modelling Language
RAML is a simple and succinct way of describing practically-RESTful APIs. See RAML.

3.3 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 Document conventions and organization

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

4.1 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 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*.

4.2 Data types

See OCF Core Specification.

5 Theory of Operation

5.1 Interworking Approach

The interworking between AllJoyn defined interfaces and OCF defined Resource Types is modelled using the derived model syntax described in Derived Models for Interoperability . Determination of the minimum set of AllJoyn interfaces for which equivalency is required within the OCF data model was done by listing the set of interfaces required for each of the device types defined by the CDM Project inside of AllJoyn. Where the AllJoyn interface supports methods then an actuation design pattern is applied.

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 Method Invocation

The invocation of a method or remote procedure call (RPC) 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 method or RPC name. The method name always includes trailing parentheses which would include any parameters should they be passed.
For example when dealing with the switchon() method from AllJoyn this gives a complete method invocation as: operation.oncontrol::switchon().

6 Device Type Mapping

6.1 Introduction

This Section contains the mappings to/from Device Types.

6.2 AllJoyn Device Types to OCF Device Types

The following table captures the equivalency mapping between AllJoyn defined Device Types (see AllJoyn Common Data Model Interface Definitions) and OCF defined Device Types (see Table 10-1 in OCF Smart Home Device Specification). The minimum interface set for the AllJoyn definitions is provided in the HAE Theory of Operation; the minimum Resource sets for each OCF Device is provided in OCF Smart Home Device Specification.

Table 6-1 AllJoyn to OCF Device Type Mapping.

Classification	AllJoyn Device Type	AllJoyn ID	OCF Device Type
Air Care	Air Conditioner	5	oic.d.airconditioner
	Air Purifier	9	oic.d.airpurifier
	Air Quality Monitor	11	oic.d.aqm
	Dehumidifier	8	oic.d.dehumidifier
	Humidifier	7	oic.d.humidifier
	Electric Fan	10	oic.d.fan
	Thermostat	6	oic.d.thermostat
Fabric Care	Clothes Washer	12	oic.d.washer
	Clothes Dryer	13	oic.d.dryer
	Clothes Washer-Dryer	14	oic.d.washerdryer
Food Preservation	Refrigerator	2	oic.d.refrigerator
	Ice-Maker	4	oic.r.icemaker (maps to Resource)
	Freezer	3	oic.d.freezer
Food Preparation	Oven	17	oic.d.oven
	Cooktop	18	oic.d.cooktop
	Cookerhood	19	oic.d.cookerhood
	Food probe	20	oic.d.foodprobe
Dish Care	Dishwasher	15	oic.d.dishwasher
Floor Care	Robot Cleaner	16	oic.d.robotcleaner
Entertainment	Television	21	oic.d.tv
	Set Top Box (STB)	22	oic.d.stb

6.3 OCF Device Types with no AllJoyn Equivalent

The following table captures the Device Types defined by OCF have no direct equivalent in AllJoyn, they shall all be mapped to an AllJoyn Device Type of 'Other' (Id of '1').

Table 6-2 OCF Device Types with no AllJoyn EquivalentOCF Device Name	OCF Device Type
Receiver	oic.d.receiver
Blind	oic.d.blind
Door	oic.d.door
Garage Door	oic.d.garagedoor
Generic Sensor	oic.d.sensor
Light	oic.d.light
Smart Plug	oic.d.smartplug
Switch	oic.d.switch
Water Valve	oic.d.watervalve
Printer	oic.d.printer
Multi-Function Printer	oic.d.multifunctionprinter
Scanner	oic.r.scanner
Camera	oic.d.camera
Security Panel	oic.d.securitypanel
Smart Lock	oic.d.smartlock

7 Resource to Interface Equivalence

7.1 Introduction

This Section lists the complete set of applicable AllJoyn Interfaces and provides the equivalent OCF Resource Type(s) to which the Interfaces map.

7.2 AllJoyn Interfaces to OCF Resources

The following tables capture the equivalency mapping between AllJoyn defined Interfaces (see AllJoyn Common Data Model Interface Definitions) and OCF defined Resource Types (see OCF Resource Type Specification). Detailed Property by Property mappings are provided in Section 8.

Table 7-1 AllJoyn Interface to OCF Resource Type Mapping – Minimum Interface Set captures the mappings for Interfaces that are part of the minimum set for an AllJoyn Device.

Table 7-2 AllJoyn Interface to OCF Resource Type Mapping – Optional Interface Set captures the mappings for Interfaces that are optional for an AllJoyn Device; deep translation for these interfaces via derived modelling is not within the scope of this release of the specification.

Table 7-1 AllJoyn Interface to OCF Resource Type Mapping – Minimum Interface Set

AllJoyn Interface	OCF Resource Type Name	OCF Resource Type ID	OCF Interface(s)
Environment.CurrentAirQuality	Air Quality Collection	oic.r.airqualitycollection	oic.if.s
Environment.CurrentAirQualityLevel	Air Quality Collection	oic.r.airqualitycollection	oic.if.s
Environment.CurrentHumidity	Humidity	oic.r.humidity	oic.if.s
Environment.CurrentTemperature	Temperature	oic.r.temperature	oic.if.s
Environment.TargetHumidity	Humidity	oic.r.humidity, oic.r.selectablelevels	oic.if.a
Environment.TargetTemperature	Temperature	oic.r.temperature	oic.if.a
Operation.AudioVolume	Audio Controls	oic.r.audio	oic.if.a
Operation.Channel	Not mapped		
Operation.ClimateControlMode	Mode	oic.r.mode	oic.if.a
	Operational State	oic.r.operational.state	oic.if.s
Operation.ClosedStatus	Door	oic.r.door	oic.if.s

Operation.CycleControl	Operational State	oic.r.operational.state	oic.if.s
Operation.FanSpeedLevel	Air Flow	oic.r.airflow	oic.if.a
Operation.HeatingZone	Heating Zone Collection	oic.r.heatingzonecollection	oic.if.s
Operation.HvacFanMode	Mode	oic.r.mode	oic.if.a
Operation.OnOffStatus	Binary Switch	oic.r.switch.binary	oic.if.s
Operation.OvenCyclePhase	Operational State	oic.r.operationalstate	oic.if.s

Table 7-2 AllJoyn Interface to OCF Resource Type Mapping – Optional Interface Set

AllJoyn Interface	OCF Resource Type Name	OCF Resource Type ID	OCF Interface(s)
Environment.TargetTemperatureLevel	Mode	oic.r.mode	oic.if.a
Environment.WaterLevel	New Resource	TBD	oic.if.s
Environment.WindDirection	Air Flow	oic.r.airflow	oic.if.a
Operation.AirRecirculationMode	Mode	oic.r.mode	oic.if.a
Operation.Alerts	TBD	TBD	TBD
Operation.AudioVideoInput	Media Source List	oic.r.media.input	oic.if.a
Operation.BatteryStatus	Battery	oic.r.energy.battery	oic.if.s
Operation.CurrentPower	Energy Usage	oic.r.energy.usage	oic.if.s
Operation.DishWashingCyclePhase	Operational State	oic.r.operationalstate	oic.if.s
Operation.EnergyUsage	Energy Usage	oic.r.energy.usage	oic.if.s

Operation.FilterStatus	New Resource	TBD	TBD
Operation.LaundryCyclePhase	Mode	oic.r.mode	oic.if.s
Operation.MoistureOutputLevel	Mode	oic.r.mode	oic.if.a
Operation.PlugInUnits	TBD	TBD	TBD
Operation.RapidMode	Refrigeration	oic.r.refrigeration	oic.if.a
Operation.RemoteControllability	TBD	TBD	TBD
Operation.RepeatMode	Ecomode	oic.r.ecomode	oic.if.a
Operation.ResourceSaving	New Resource	TBD	TBD
Operation.RobotCleaningCyclePhase	Mode	oic.r.mode	oic.if.s
Operation.SoilLevel	Mode	oic.r.mode	oic.if.a
Operation.SpinSpeedLevel	Mode	oic.r.mode	oic.if.a
Operation.Timer	Time Period	oic.r.time.period	oic.if.s

8 Detailed Mapping APIs

8.1 Introduction

This section provides an API (using RAML) and a mapping description (using JSON that aligns with the Derived Modeling syntax described in [Derived Model White Paper]) for all Interfaces and Resources that are within scope.

Annex A provides definitions for all mappings making use of Swagger2.0 instead of RAML and JSON.

All the sub-clauses in clause 8 and Annex A describe the Resource Types with a restful API definition language. The Resource Type definitions presented in clause 8 and Annex A are formatted for readability, and so may appear to have extra line breaks. The contents of the Resource Types without the extra line breaks are available in OCF Resource Type Definitions.

Table 8-1 Interface to Resource Summary provides a reference and link to the per Interface sub-sections.

Table 8-1 Interface to Resource Summary

AllJoyn Interface Name	Equivalent Resource(s)	Mapping Section
Environment.CurrentAirQuality	oic.r.airqualitycollection	8.2
Environment.CurrentAirQualityLevel	oic.r.airqualitycollection	8.3
Environment.CurrentHumidity	oic.r.humidity	8.4
Environment.CurrentTemperature	oic.r.temperature	8.5
Environment.TargetHumidity	oic.r.humidity, oic.r.selectablelevels	8.6
Environment.TargetTemperature	oic.r.temperature	8.7
Operation.AudioVolume	oic.r.audio	8.8
Operation.ClimateControlMode	oic.r.mode, oic.r.operationalstate	8.9
Operation.ClosedStatus	oic.r.door	8.10
Operation.CycleControl	oic.r.operational.state	8.11
Operation.FanSpeedLevel	oic.r.airflow	8.12
Operation.HeatingZone	oic.r.heatingzonecollection	8.13
Operation.HvacFanMode	oic.r.mode	8.14
Operation.OnOffStatus, Operation.OnControl, Operation.OffControl	oic.r.switch.binary	8.15
Operation.OvenCyclePhase	oic.r.operationalstate	8.16

8.2 Air Quality Mapping

8.2.1 Introduction

This API defines the mapping between the AllJoyn AirQuality interface and the OCF AirQuality Resource. If more than one instance of the AirQuality interface is exposed then each instance maps to an instance of the OCF AirQuality Resource. The mapping defined in the schema describes the population of the OCF AirQuality Resource. Even if there is only a single instance of an OCF AirQuality Resource this shall be included in an instance of an OCF AirQualityCollection. The number of links in the collection equates to the number of instances of the AllJoyn CurrentAirQuality interface that are exposed. When mapping from OCF the valueType of the Resource shall be introspected, this API is invoked only if this is set to 'Measured'

8.2.2 Example URI

/CurrentAirQualityResURI

8.2.3 Resource Type

The resource type (rt) is defined as: oic.r.airqualitycollection.

8.2.4 RAML Definition

```
#%RAML 0.8

title: CurrentAirQualityInterfaceMapping
version: OCFv1.0.0-20170317

traits:
 - interface-sensor :
     queryParameters:

       if:
         enum: ["oic.if.s", "oic.if.baseline"]


/CurrentAirQualityResURI:

  description: |
    This API defines the mapping between the AllJoyn AirQuality interface and the OCF AirQuality
Resource.
    If more than one instance of the AirQuality interface is exposed then each instance maps to an
instance of the OCF AirQuality Resource.
    The mapping defined in the schema describes the population of the OCF AirQuality Resource.
    Even if there is only a single instance of an OCF AirQuality Resource this shall be included in
an instance of an OCF AirQualityCollection.
    The number of links in the collection equates to the number of instances of the AllJoyn
CurrentAirQuality interface that are exposed.
    When mapping from OCF the valueType of the Resource shall be introspected, this API is invoked
only if this is set to 'Measured'


  is : ['interface-sensor']

  get:
    responses :

      200:

        body:
          application/json:

            schema: |

              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.environment.currentairquality.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Current Air Quality",
                "definitions": {
                  "asa.environment.currentairquality": {
                    "type": "object",
                    "properties": {
                      "contaminanttype": {
                        "type": "integer",
                        "description": "The contaminant type",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.airquality",
                          "x-to-ocf": [
                            "valuetype = Measured",
                            "contaminanttypearray = [CH2O,CO2,CO,PM2_5,PM10,VOC]",
                            "ocf.contaminanttype = contaminanttypearray[contaminanttype]"
                          ],
                          "x-from-ocf": [
                            "contaminanttype = indexof contaminanttypearray[ocf.contaminanttype]"
                          ]
                        }
```

```
        },
        "currentvalue": {
          "type": "number",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "contaminantvalue = currentvalue"
            ],
            "x-from-ocf": [
              "currentvalue = contaminantvalue"
            ]
          }
        },
        "minvalue": {
          "type": "number",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "range[0] = minvalue"
            ],
            "x-from-ocf": [
              "minvalue = range[0]"
            ]
          }
        },
        "maxvalue": {
          "type": "number",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "range[1] = maxvalue"
            ],
            "x-from-ocf": [
              "maxvalue = range[1]"
            ]
          }
        },
        "precision": {
          "type": "number",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "ocf.precision = precision"
            ],
            "x-from-ocf": [
              "precision = ocf.precision"
            ]
          }
        },
        "updatemintime": {
          "type": "integer",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.value.conditional",
            "x-to-ocf": [
              "ocf.minnotifyperiod = updatemintime"
            ],
            "x-from-ocf": [
              "updatemintime = ocf.minnotifyperiod"
            ]
          }
        }
      }
    },
    "type": "object",
    "allOf": [
      {"$ref": "#/definitions/asa.environment.currentairquality"}
    ],
    "required":
["contaminanttype","currentvalue","minvalue","maxvalue","precision","updatemintime"]
```

```
  }

example: |
  {
    "rt":     ["oic.r.airqualitycollection"]
  }
```

8.2.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
currentvalue	oic.r.airquality	contaminantvalue = currentvalue	currentvalue = contaminantvalue	
updatemintime	oic.r.value.conditional	ocf.minnotifyperiod = updatemintime	updatemintime = ocf.minnotifyperiod	
maxvalue	oic.r.airquality	range[1] = maxvalue	maxvalue = range[1]	
precision	oic.r.airquality	ocf.precision = precision	precision = ocf.precision	
minvalue	oic.r.airquality	range[0] = minvalue	minvalue = range[0]	
contaminanttype	oic.r.airquality	valuetype = Measuredcontaminanttypearray = [CH2O,CO2,CO,PM2_5,PM10,VOC]ocf.contaminanttype = contaminanttypearray[contaminanttype]	contaminanttype = indexof contaminanttypearray[ocf.contaminanttype]	The contaminant type

8.2.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentAirQualityResURI		get			

8.3 Air Quality Level Mapping

8.3.1 Introduction

This API defines the mapping between the AllJoyn AirQualityLevel interface and the OCF AirQuality Resource. If more than one instance of the AirQualityLevel interface is exposed then each instance maps to an instance of the OCF AirQuality Resource. The mapping defined in the schema describes the population of the OCF AirQuality Resource. Even if there is only a single instance of an OCF AirQuality Resource then this shall be included in an instance of an OCF AirQualityCollection. The number of links in the collection equates to the number of instances of the AllJoyn CurrentAirQuality interface that are exposed. When mapping from OCF the valueType of the Resource shall be introspected, this API is invoked only if this is set to 'Qualitative'

8.3.2 Example URI

/CurrentAirQualityLevelResURI

8.3.3 Resource Type

The resource type (rt) is defined as: oic.r.airqualitycollection.

8.3.4 RAML Definition

```
#%RAML 0.8

title: CurrentAirQualityLevelInterfaceMapping
version: OCFv1.0.0-20170317
```

```
traits:
 - interface-sensor :
     queryParameters:

       if:
         enum: ["oic.if.s", "oic.if.baseline"]


/CurrentAirQualityLevelResURI:

  description: |
    This API defines the mapping between the AllJoyn AirQualityLevel interface and the OCF
AirQuality Resource.
    If more than one instance of the AirQualityLevel interface is exposed then each instance maps
to an instance of the OCF AirQuality Resource.
    The mapping defined in the schema describes the population of the OCF AirQuality Resource.
    Even if there is only a single instance of an OCF AirQuality Resource then this shall be
included in an instance of an OCF AirQualityCollection.
    The number of links in the collection equates to the number of instances of the AllJoyn
CurrentAirQuality interface that are exposed.
    When mapping from OCF the valueType of the Resource shall be introspected, this API is invoked
only if this is set to 'Qualitative'


  is : ['interface-sensor']

  get:
    responses :

      200:

        body:
          application/json:

            schema: |

              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.environment.currentairqualitylevel.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Current Air Quality Level",
                "definitions": {
                  "asa.environment.currentairqualitylevel": {
                    "type": "object",
                    "properties": {
                      "contaminanttype": {
                        "type": "integer",
                        "description": "The contaminant type",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.airquality",
                          "x-to-ocf": [
                            "valuetype = Qualitative",
                            "if contaminanttype = 0, ocf.contaminanttype = CH2O",
                            "if contaminanttype = 1, ocf.contaminanttype = CO2",
                            "if contaminanttype = 2, ocf.contaminanttype = CO",
                            "if contaminanttype = 3, ocf.contaminanttype = PM2_5",
                            "if contaminanttype = 4, ocf.contaminanttype = PM10",
                            "if contaminanttype = 5, ocf.contaminanttype = VOC",
                            "if contaminanttype = 253, ocf.contaminanttype = Smoke",
                            "if contaminanttype = 254, ocf.contaminanttype = Odor",
                            "if contaminanttype = 255, ocf.contaminanttype = AirPollution"
                          ],
                          "x-from-ocf": [
                            "if ocf.contaminanttype = CH2O, contaminanttype = 0",
                            "if ocf.contaminanttype = CO2, contaminanttype = 1",
                            "if ocf.contaminanttype = CO, contaminanttype = 2",
                            "if ocf.contaminanttype = PM2_5, contaminanttype = 3",
                            "if ocf.contaminanttype = PM10, contaminanttype = 4",
                            "if ocf.contaminanttype = VOC, contaminanttype = 5",
                            "if ocf.contaminanttype = Smoke, contaminanttype = 253",
                            "if ocf.contaminanttype = Odor, contaminanttype = 254",
                            "if ocf.contaminanttype = AirPollution, contaminanttype = 255"
```

```
            ]
          }
        },
        "currentlevel": {
          "type": "integer",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "contaminantvalue = currentlevel"
            ],
            "x-from-ocf": [
              "currentlevel = contaminantvalue"
            ]
          }
        },
        "maxlevel": {
          "type": "integer",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "range[0] = 0",
              "range[1] = maxvalue"
            ],
            "x-from-ocf": [
              "maxvalue = range[1]"
            ]
          }
        }
      }
    }
  },
  "type": "object",
  "allOf": [
    {"$ref": "#/definitions/asa.environment.currentairqualitylevel"}
  ],
  "required": ["contaminanttype","currentlevel","maxlevel"]
}

example: |
  {
    "rt":     ["oic.r.airqualitycollection"]
  }
```

8.3.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
currentlevel	oic.r.airquality	contaminantvalue = currentlevel	currentlevel = contaminantvalue	
maxlevel	oic.r.airquality	range[0] = 0range[1] = maxvalue	maxvalue = range[1]	
contaminanttype	oic.r.airquality	valuetype = Qualitativeif contaminanttype = 0, ocf.contaminanttype = CH2Oif contaminanttype = 1, ocf.contaminanttype = CO2if contaminanttype = 2, ocf.contaminanttype	if ocf.contaminanttype = CH2O, contaminanttype = 0if ocf.contaminanttype = CO2, contaminanttype = 1if ocf.contaminanttype = CO, contaminanttype = 2if	The contaminant type

		= COif contaminanttype = 3, ocf.contaminanttype = PM2_5if contaminanttype = 4, ocf.contaminanttype = PM10if contaminanttype = 5, ocf.contaminanttype = VOCif contaminanttype = 253, ocf.contaminanttype = Smokeif contaminanttype = 254, ocf.contaminanttype = Odorif contaminanttype = 255, ocf.contaminanttype = AirPollution	ocf.contaminanttype = PM2_5, contaminanttype = 3if ocf.contaminanttype = PM10, contaminanttype = 4if ocf.contaminanttype = VOC, contaminanttype = 5if ocf.contaminanttype = Smoke, contaminanttype = 253if ocf.contaminanttype = Odor, contaminanttype = 254if ocf.contaminanttype = AirPollution, contaminanttype = 255	

8.3.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentAirQualityLevelResURI		get			

8.4 Current Humidity Mapping

8.4.1 Introduction

This API defines the mapping between an instance of an OCF Humidity which exposes only a sensor interface and the AllJoyn Current Humidity interface. A RETRIEVE on a Temperature Sensor maps to an action on an instance of an Environment.CurrentTemperature Interface.

8.4.2 Example URI

/CurrentHumidityResURI

8.4.3 Resource Type

The resource type (rt) is defined as: oic.r.humidity.

8.4.4 RAML Definition

```
#%RAML 0.8
title: CurrentHumidityInterfaceMapping
version: OCFv1.0.0-20170317
traits:
 - interface-sensor :
     queryParameters:
       if:
         enum: ["oic.if.s", "oic.if.baseline"]


/CurrentHumidityResURI:
  description: |
    This API defines the mapping between an instance of an OCF Humidity which exposes only a sensor
interface
```

```
    and the AllJoyn Current Humidity interface.
    A RETRIEVE on a Temperature Sensor maps to an action on an instance of an
Environment.CurrentTemperature Interface.

  is : ['interface-sensor']
  get:
    responses :
      200:
        body:
          application/json:
            schema: |
              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.environment.currenthumidity.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Current Humidity",
                "definitions": {
                  "asa.environment.currenthumidity": {
                    "type": "object",
                    "properties": {
                      "currentvalue": {
                        "type": "number",
                        "description": "Measured value",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.humidity",
                          "x-to-ocf": [
                            "humidity = currentValue"
                          ],
                          "x-from-ocf": [
                            "currentvalue = humidity"
                          ]
                        }
                      },
                      "maxvalue": {
                        "type": "number",
                        "description": "Max measured value for humidty",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.humidity",
                          "x-to-ocf": [
                            "range[0] = 0",
                            "range[1] = maxvalue"
                          ],
                          "x-from-ocf": [
                            "maxvalue = range[1]"
                          ]
                        }
                      }
                    }
                  },
                  "type": "object",
                  "allOf": [
                    {"$ref": "#/definitions/asa.environment.currenthumidity"}
                  ],
                  "required": [ "currentvalue","maxvalue"]
                }

            example: |
              {
                "rt":     ["oic.r.humidity"]
              }
```

8.4.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
currentvalue	oic.r.humidity	humidity = currentValue	currentvalue = humidity	Measured value
maxvalue	oic.r.humidity	range[0] = 0range[1] = maxvalue	maxvalue = range[1]	Max measured value for humidty

8.4.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentHumidityResURI		get			

8.5 Current Temperature Mapping

8.5.1 Introduction

This API defines the mapping between an instance of an OCF Temperature which exposes only a sensor interface and the AllJoyn Current Temperature interface. A RETRIEVE on a Temperature Sensor maps to an action on an instance of an Environment.CurrentTemperature Interface.

8.5.2 Example URI

/CurrentTemperatureResURI

8.5.3 Resource Type

The resource type (rt) is defined as: oic.r.temperature.

8.5.4 RAML Definition

```
#%RAML 0.8

title: CurrentTemperatureInterfaceMapping
version: OCFv1.0.0-20170317

traits:
 - interface-sensor :
     queryParameters:

       if:
         enum: ["oic.if.s", "oic.if.baseline"]


/CurrentTemperatureResURI:

  description: |
    This API defines the mapping between an instance of an OCF Temperature which exposes only a
sensor interface
    and the AllJoyn Current Temperature interface.
    A RETRIEVE on a Temperature Sensor maps to an action on an instance of an
Environment.CurrentTemperature Interface.


  is : ['interface-sensor']

  get:
    responses :

      200:

        body:
          application/json:

            schema: |

              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.environment.currenttemperature.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
```

```
"title": "Current Temperature",
"definitions": {
  "asa.environment.currenttemperature": {
    "type": "object",
    "properties": {
      "currentvalue": {
        "type": "number",
        "description": "Measured value",
        "x-ocf-conversion": {
          "x-ocf-alias": "oic.r.temperature",
          "x-to-ocf": [
            "temperature = currentValue",
            "units = C"
          ],
          "x-from-ocf": {
            "oneOf": [
              {
                "properties": {
                  "units": "string",
                  "enum": ["C"]
                },
                "x-from-ocf": [
                  "currentvalue = temperature"
                ]
              },
              {
                "properties": {
                  "units": "string",
                  "enum": ["F"]
                },
                "x-from-ocf": [
                  "currentvalue = (temperature-32)*5/9"
                ]
              },
              {
                "properties": {
                  "units": "string",
                  "enum": ["K"]
                },
                "x-from-ocf": [
                  "currentvalue = temperature-273.15"
                ]
              }
            ]
          }
        }
      },
      "precision": {
        "type": "number",
        "x-ocf-conversion": {
          "x-ocf-alias": "oic.r.temperature",
          "x-to-ocf": [
            "ocf.precision = precision"
          ],
          "x-from-ocf": [
            "precision = ocf.precision"
          ]
        }
      },
      "updatemintime": {
        "type": "integer",
        "x-ocf-conversion": {
          "x-ocf-alias": "oic.r.value.conditional",
          "x-to-ocf": [
            "ocf.minnotifyperiod = updatemintime"
          ],
          "x-from-ocf": [
            "updatemintime = ocf.minnotifyperiod"
          ]
        }
      }
```

```
        }
      }
    },
    "type": "object",
    "allOf": [
      {"$ref": "#/definitions/asa.environment.currenttemperature"}
    ],
    "required": [ "currentvalue","precision","updatemintime" ]
  }

example: |
  {
    "rt":     ["oic.r.temperature"]
  }
```

8.5.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
currentvalue	oic.r.temperature	temperature = currentValueunits = C	oneOf	Measured value
updatemintime	oic.r.value.conditional	ocf.minnotifyperiod = updatemintime	updatemintime = ocf.minnotifyperiod	
precision	oic.r.temperature	ocf.precision = precision	precision = ocf.precision	

8.5.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentTemperatureResURI		get			

8.6 Target Humidity Mapping

8.6.1 Introduction

This API defines the mapping between an instance of an AllJoyn TargetHumidity Interface and the OCF Resource Equivalent. A POST on a Humidity Sensor maps to an action on an instance of an Environment.TargetHumidity Interface.

8.6.2 Example URI

/TargetHumidityResURI

8.6.3 Resource Type

The resource type (rt) is defined as: oic.r.humidity.

8.6.4 RAML Definition

```
#%RAML 0.8
title: TargetHumidityInterfaceMapping
version: OCFv1.0.0-20170317

traits:
 - interface-actuator :
     queryParameters:

       if:
         enum: ["oic.if.a", "oic.if.baseline"]


/TargetHumidityResURI:
  description: |
```

```
    This API defines the mapping between an instance of an AllJoyn TargetHumidity Interface and the
OCF Resource Equivalent.
    A POST on a Humidity Sensor maps to an action on an instance of an Environment.TargetHumidity
Interface.

  is : ['interface-actuator']

  get:
    responses :

      200:

        body:
          application/json:

            schema: |

              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.environment.targethumidity.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Target Humidity",
                "definitions": {
                  "asa.environment.targethumidity": {
                    "type": "object",
                    "properties": {
                      "targetvalue": {
                        "type": "number",
                        "description": "Measured value",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
                          "x-to-ocf": [
                            "if minvalue != maxvalue, ocf.desiredhumidity =
targetvalue;ocf.targetlevel = selectablehumiditylevels[0].",
                            "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
                          ],
                          "x-from-ocf": [
                            "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
                            "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
                          ]
                        }
                      },
                      "minvalue": {
                        "type": "number",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.humidity",
                          "x-to-ocf": [
                            "range[0] = minvalue"
                          ],
                          "x-from-ocf": [
                            "minvalue = range[0]",
                            "otherwise: minvalue = 0"
                          ]
                        }
                      },
                      "maxvalue": {
                        "type": "number",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.humidity",
                          "x-to-ocf": [
                            "range[1] = maxvalue"
                          ],
                          "x-from-ocf": [
                            "maxvalue = range[1]",
                            "otherwise: maxvalue = 100"
                          ]
                        }
                      },
                      "stepvalue": {
                        "type": "number",
                        "x-ocf-conversion": {
```

```
                        "x-ocf-alias": "oic.r.humidity",
                        "x-to-ocf": [
                          "step = stepvalue"
                        ],
                        "x-from-ocf": [
                          "stepvalue = step",
                          "otherwise: stepvalue = 1"
                        ]
                      }
                    },
                    "selectablehumiditylevels": {
                      "type": "array",
                      "items": {
                        "type": "number"
                      },
                      "x-ocf-conversion": {
                        "x-ocf-alias": "oic.r.selectablelevels",
                        "x-to-ocf": [
                          "availablelevels[] = selectablehumiditylevels[]"
                        ],
                        "x-from-ocf": [
                          "selectablehumiditylevels[] = availablelevels[]"
                        ]
                      }
                    }
                  }
                },
                "type": "object",
                "allOf": [
                  {"$ref": "#/definitions/asa.environment.targethumidity"}
                ],
                "required":
[ "targetvalue","minvalue","maxvalue","stepvalue","selectablehumiditylevels" ]
              }

            example: |
              {
                "rt":     ["oic.r.humidity","oic.r.selectablelevels"]
              }


  post:
    body:
      application/json:

        schema: |
          {
            "id":
"http://openinterconnect.org/asamapping/schemas/asa.environment.targethumidity.json#",
            "$schema": "http://json-schema.org/draft-04/schema#",
            "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
            "title": "Target Humidity",
            "definitions": {
              "asa.environment.targethumidity": {
                "type": "object",
                "properties": {
                  "targetvalue": {
                    "type": "number",
                    "description": "Measured value",
                    "x-ocf-conversion": {
                      "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
                      "x-to-ocf": [
                        "if minvalue != maxvalue, ocf.desiredhumidity = targetvalue;ocf.targetlevel
= selectablehumiditylevels[0].",
                        "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
                      ],
                      "x-from-ocf": [
                        "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
```

```
                    "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
                  ]
                }
              },
              "minvalue": {
                "type": "number",
                "x-ocf-conversion": {
                  "x-ocf-alias": "oic.r.humidity",
                  "x-to-ocf": [
                    "range[0] = minvalue"
                  ],
                  "x-from-ocf": [
                    "minvalue = range[0]",
                    "otherwise: minvalue = 0"
                  ]
                }
              },
              "maxvalue": {
                "type": "number",
                "x-ocf-conversion": {
                  "x-ocf-alias": "oic.r.humidity",
                  "x-to-ocf": [
                    "range[1] = maxvalue"
                  ],
                  "x-from-ocf": [
                    "maxvalue = range[1]",
                    "otherwise: maxvalue = 100"
                  ]
                }
              },
              "stepvalue": {
                "type": "number",
                "x-ocf-conversion": {
                  "x-ocf-alias": "oic.r.humidity",
                  "x-to-ocf": [
                    "step = stepvalue"
                  ],
                  "x-from-ocf": [
                    "stepvalue = step",
                    "otherwise: stepvalue = 1"
                  ]
                }
              },
              "selectablehumiditylevels": {
                "type": "array",
                "items": {
                  "type": "number"
                },
                "x-ocf-conversion": {
                  "x-ocf-alias": "oic.r.selectablelevels",
                  "x-to-ocf": [
                    "availablelevels[] = selectablehumiditylevels[]"
                  ],
                  "x-from-ocf": [
                    "selectablehumiditylevels[] = availablelevels[]"
                  ]
                }
              }
            }
          }
        },
        "type": "object",
        "allOf": [
          {"$ref": "#/definitions/asa.environment.targethumidity"}
        ],
        "required":
[ "targetvalue","minvalue","maxvalue","stepvalue","selectablehumiditylevels" ]
      }
```

responses :

```
      200:
        body:
          application/json:
            schema: |
              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.environment.targethumidity.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Target Humidity",
                "definitions": {
                  "asa.environment.targethumidity": {
                    "type": "object",
                    "properties": {
                      "targetvalue": {
                        "type": "number",
                        "description": "Measured value",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
                          "x-to-ocf": [
                            "if minvalue != maxvalue, ocf.desiredhumidity =
targetvalue;ocf.targetlevel = selectablehumiditylevels[0].",
                            "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
                          ],
                          "x-from-ocf": [
                            "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
                            "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
                          ]
                        }
                      },
                      "minvalue": {
                        "type": "number",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.humidity",
                          "x-to-ocf": [
                            "range[0] = minvalue"
                          ],
                          "x-from-ocf": [
                            "minvalue = range[0]",
                            "otherwise: minvalue = 0"
                          ]
                        }
                      },
                      "maxvalue": {
                        "type": "number",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.humidity",
                          "x-to-ocf": [
                            "range[1] = maxvalue"
                          ],
                          "x-from-ocf": [
                            "maxvalue = range[1]",
                            "otherwise: maxvalue = 100"
                          ]
                        }
                      },
                      "stepvalue": {
                        "type": "number",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.humidity",
                          "x-to-ocf": [
                            "step = stepvalue"
                          ],
                          "x-from-ocf": [
                            "stepvalue = step",
                            "otherwise: stepvalue = 1"
                          ]
                        }
```

```
                    },
                    "selectablehumiditylevels": {
                      "type": "array",
                      "items": {
                        "type": "number"
                      },
                      "x-ocf-conversion": {
                        "x-ocf-alias": "oic.r.selectablelevels",
                        "x-to-ocf": [
                          "availablelevels[] = selectablehumiditylevels[]"
                        ],
                        "x-from-ocf": [
                          "selectablehumiditylevels[] = availablelevels[]"
                        ]
                      }
                    }
                  }
                },
                "type": "object",
                "allOf": [
                  {"$ref": "#/definitions/asa.environment.targethumidity"}
                ],
                "required":
[ "targetvalue","minvalue","maxvalue","stepvalue","selectablehumiditylevels" ]
              }

            example: |
              {
                "rt":     ["oic.r.humidity","oic.r.selectablelevels"]
              }
```

8.6.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
stepvalue	oic.r.humidity	step = stepvalue	stepvalue = stepotherwise: stepvalue = 1	
targetvalue	oic.r.humidity,oic.r.selectablelevels	if minvalue != maxvalue, ocf.desiredhumidity = targetvalue;ocf.targetlevel = selectablehumiditylevels[0].if minvalue == maxvalue, ocf.targetlevel = targetvalue.	if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel.	Measured value
maxvalue	oic.r.humidity	range[1] = maxvalue	maxvalue = range[1]otherwise: maxvalue = 100	
selectablehumiditylevels	oic.r.selectablelevels	availablelevels[] = selectablehumiditylevels[]	selectablehumiditylevels[] = availablelevels[]	
minvalue	oic.r.humidity	range[0] = minvalue	minvalue = range[0]otherwise: minvalue = 0	

8.6.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/TargetHumidityResURI		get	post		

8.7 Target Temperature Mapping

8.7.1 Introduction

This API defines the mapping between an instance of an OCF Temperature which exposes only a sensor interface and the AllJoyn Current Temperature interface. A RETRIEVE on a Temperature Sensor maps to an action on an instance of an Environment.CurrentTemperature Interface.

8.7.2 Example URI

/TargetTemperatureResURI

8.7.3 Resource Type

The resource type (rt) is defined as: oic.r.temperature.

8.7.4 RAML Definition

```
#%RAML 0.8

title: TargetTemperatureInterfaceMapping
version: OCFv1.0.0-20170317

traits:
 - interface-actuator :
     queryParameters:

       if:
         enum: ["oic.if.a", "oic.if.baseline"]


/TargetTemperatureResURI:

  description: |
    This API defines the mapping between an instance of an OCF Temperature which exposes only a
sensor interface
    and the AllJoyn Current Temperature interface.
    A RETRIEVE on a Temperature Sensor maps to an action on an instance of an
Environment.CurrentTemperature Interface.


  is : ['interface-actuator']

  get:
    responses :

      200:

        body:
          application/json:

            schema:

              |
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.environment.targettemperature.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Target Temperature",
                "definitions": {
                  "asa.environment.targettemperature": {
                    "type": "object",
                    "properties": {
                      "targetvalue": {
                        "type": "number",
                        "description": "Measured value",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.temperature",
                          "x-to-ocf": [
                            "temperature = targetvalue",
```

```
        "units = C"
      ],
      "x-from-ocf": {
        "oneOf": [
          {
            "properties": {
              "units": "string",
              "enum": ["C"]
            },
            "x-from-ocf": [
              "targetvalue = temperature"
            ]
          },
          {
            "properties": {
              "units": "string",
              "enum": ["F"]
            },
            "x-from-ocf": [
              "targetvalue = (temperature-32)*5/9"
            ]
          },
          {
            "properties": {
              "units": "string",
              "enum": ["K"]
            },
            "x-from-ocf": [
              "targetvalue = temperature-273.15"
            ]
          }
        ]
      }
    }
  },
  "minvalue": {
    "type": "number",
    "x-ocf-conversion": {
      "x-ocf-alias": "oic.r.temperature",
      "x-to-ocf": [
        "range[0] = minvalue"
      ],
      "x-from-ocf": [
        "minvalue = range[0]",
        "otherwise: minvalue = -MAXINT"
      ]
    }
  },
  "maxvalue": {
    "type": "number",
    "x-ocf-conversion": {
      "x-ocf-alias": "oic.r.temperature",
      "x-to-ocf": [
        "range[1] = maxvalue"
      ],
      "x-from-ocf": [
        "maxvalue = range[1]",
        "otherwise: maxvalue = MAXINT"
      ]
    }
  },
  "step": {
    "type": "number",
    "x-ocf-conversion": {
      "x-ocf-alias": "oic.r.temperature",
      "x-to-ocf": [
        "ocf.step = step"
      ],
      "x-from-ocf": [
        "step = ocf.step",
        "otherwise: step = undefined (0x00)"
```

```
                        ]
                      }
                    }
                  }
                }
              },
              "type": "object",
              "allOf": [
                {"$ref": "#/definitions/asa.environment.targettemperature"}
              ],
              "required": [ "targetvalue","minvalue","maxvalue","step" ]
            }

          example: |
            {
              "rt":     ["oic.r.temperature"]
            }

  post:
    body:
      application/json:
        schema: |
          {
            "id":
"http://openinterconnect.org/asamapping/schemas/asa.environment.targettemperature.json#",
            "$schema": "http://json-schema.org/draft-04/schema#",
            "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
            "title": "Target Temperature",
            "definitions": {
              "asa.environment.targettemperature": {
                "type": "object",
                "properties": {
                  "targetvalue": {
                    "type": "number",
                    "description": "Measured value",
                    "x-ocf-conversion": {
                      "x-ocf-alias": "oic.r.temperature",
                      "x-to-ocf": [
                        "temperature = targetvalue",
                        "units = C"
                      ],
                      "x-from-ocf": {
                        "oneOf": [
                          {
                            "properties": {
                              "units": "string",
                              "enum": ["C"]
                            },
                            "x-from-ocf": [
                              "targetvalue = temperature"
                            ]
                          },
                          {
                            "properties": {
                              "units": "string",
                              "enum": ["F"]
                            },
                            "x-from-ocf": [
                              "targetvalue = (temperature-32)*5/9"
                            ]
                          },
                          {
                            "properties": {
                              "units": "string",
                              "enum": ["K"]
                            },
                            "x-from-ocf": [
```

```
                        "targetvalue = temperature-273.15"
                      ]
                    }
                  ]
                }
              }
            },
            "minvalue": {
              "type": "number",
              "x-ocf-conversion": {
                "x-ocf-alias": "oic.r.temperature",
                "x-to-ocf": [
                  "range[0] = minvalue"
                ],
                "x-from-ocf": [
                  "minvalue = range[0]",
                  "otherwise: minvalue = -MAXINT"
                ]
              }
            },
            "maxvalue": {
              "type": "number",
              "x-ocf-conversion": {
                "x-ocf-alias": "oic.r.temperature",
                "x-to-ocf": [
                  "range[1] = maxvalue"
                ],
                "x-from-ocf": [
                  "maxvalue = range[1]",
                  "otherwise: maxvalue = MAXINT"
                ]
              }
            },
            "step": {
              "type": "number",
              "x-ocf-conversion": {
                "x-ocf-alias": "oic.r.temperature",
                "x-to-ocf": [
                  "ocf.step = step"
                ],
                "x-from-ocf": [
                  "step = ocf.step",
                  "otherwise: step = undefined (0x00)"
                ]
              }
            }
          }
        },
        "type": "object",
        "allOf": [
          {"$ref": "#/definitions/asa.environment.targettemperature"}
        ],
        "required": [ "targetvalue","minvalue","maxvalue","step" ]
      }

  responses :
    200:

      body:
        application/json:

          schema: |

            {
              "id":
"http://openinterconnect.org/asamapping/schemas/asa.environment.targettemperature.json#",
              "$schema": "http://json-schema.org/draft-04/schema#",
              "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
              "title": "Target Temperature",
```

```
"definitions": {
  "asa.environment.targettemperature": {
    "type": "object",
    "properties": {
      "targetvalue": {
        "type": "number",
        "description": "Measured value",
        "x-ocf-conversion": {
          "x-ocf-alias": "oic.r.temperature",
          "x-to-ocf": [
            "temperature = targetvalue",
            "units = C"
          ],
          "x-from-ocf": {
            "oneOf": [
              {
                "properties": {
                  "units": "string",
                  "enum": ["C"]
                },
                "x-from-ocf": [
                  "targetvalue = temperature"
                ]
              },
              {
                "properties": {
                  "units": "string",
                  "enum": ["F"]
                },
                "x-from-ocf": [
                  "targetvalue = (temperature-32)*5/9"
                ]
              },
              {
                "properties": {
                  "units": "string",
                  "enum": ["K"]
                },
                "x-from-ocf": [
                  "targetvalue = temperature-273.15"
                ]
              }
            ]
          }
        }
      },
      "minvalue": {
        "type": "number",
        "x-ocf-conversion": {
          "x-ocf-alias": "oic.r.temperature",
          "x-to-ocf": [
            "range[0] = minvalue"
          ],
          "x-from-ocf": [
            "minvalue = range[0]",
            "otherwise: minvalue = -MAXINT"
          ]
        }
      },
      "maxvalue": {
        "type": "number",
        "x-ocf-conversion": {
          "x-ocf-alias": "oic.r.temperature",
          "x-to-ocf": [
            "range[1] = maxvalue"
          ],
          "x-from-ocf": [
            "maxvalue = range[1]",
            "otherwise: maxvalue = MAXINT"
          ]
        }
```

```
        },
        "step": {
          "type": "number",
          "x-ocf-conversion": {
            "x-ocf-alias": "oic.r.temperature",
            "x-to-ocf": [
              "ocf.step = step"
            ],
            "x-from-ocf": [
              "step = ocf.step",
              "otherwise: step = undefined (0x00)"
            ]
          }
        }
      }
    }
  },
  "type": "object",
  "allOf": [
    {"$ref": "#/definitions/asa.environment.targettemperature"}
  ],
  "required": [ "targetvalue","minvalue","maxvalue","step" ]
}
```

8.7.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
targetvalue	oic.r.temperature	temperature = targetvalueunits = C	oneOf	Measured value
step	oic.r.temperature	ocf.step = step	step = ocf.stepotherwise: step = undefined (0x00)	
maxvalue	oic.r.temperature	range[1] = maxvalue	maxvalue = range[1]otherwise: maxvalue = MAXINT	
minvalue	oic.r.temperature	range[0] = minvalue	minvalue = range[0]otherwise: minvalue = -MAXINT	

8.7.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/TargetTemperatureResURI		get	post		

8.8 Audio Volume Mapping

8.8.1 Introduction

This API defines the mapping between an instance of an OCF Audio Controls and the AllJoyn Audio Volume interface.

8.8.2 Example URI

/AudioVolumeResURI

8.8.3 Resource Type

The resource type (rt) is defined as: oic.r.audio.

8.8.4 RAML Definition

```
#%RAML 0.8
```

```
title: AudioVolumeInterfaceMapping
version: OCFv1.0.0-20170317

traits:
 - interface-all :
     queryParameters:

       if:
         enum: ["oic.if.a", "oic.if.baseline"]


/AudioVolumeResURI:

  description: |
    This API defines the mapping between an instance of an OCF Audio Controls
    and the AllJoyn Audio Volume interface.


  is : ['interface-all']

  get:
    responses :

      200:

        body:
          application/json:

            schema: |

              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.operation.audiovolume.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Audio Volume",
                "definitions": {
                  "asa.operation.audiovolume": {
                    "type": "object",
                    "properties": {
                      "volume": {
                        "type": "integer",
                        "description": "Speaker volume index",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.audio",
                          "x-to-ocf": [
                            "ocf.volume = volume"
                          ],
                          "x-from-ocf": [
                            "volume = ocf.volume"
                          ]
                        }
                      },
                      "maxvolume": {
                        "type": "integer",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.audio",
                          "x-to-ocf": [
                            "range[0] = 0",
                            "range[1] = maxvolume"
                          ],
                          "x-from-ocf": [
                            "maxvolume = range[1]",
                            "otherwise: maxvalue = 100"
                          ]
                        }
                      },
                      "mute": {
                        "type": "boolean",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.audio",
                          "x-to-ocf": [
                            "ocf.mute = mute"
                          ],
```

```
                    "x-from-ocf": [
                      "mute = ocf.mute"
                    ]
                  }
                }
              }
            }
          },
          "type": "object",
          "allOf": [
            {"$ref": "#/definitions/asa.operation.audiovolume"}
          ],
          "required": [ "volume","maxvolume","mute" ]
        }

      example: |

        {
          "rt":     ["oic.r.audio"]
        }


post:
  body:
    application/json:

      schema: |

        {
          "id": "http://openinterconnect.org/asamapping/schemas/asa.operation.audiovolume.json#",
          "$schema": "http://json-schema.org/draft-04/schema#",
          "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
          "title": "Audio Volume",
          "definitions": {
            "asa.operation.audiovolume": {
              "type": "object",
              "properties": {
                "volume": {
                  "type": "integer",
                  "description": "Speaker volume index",
                  "x-ocf-conversion": {
                    "x-ocf-alias": "oic.r.audio",
                    "x-to-ocf": [
                      "ocf.volume = volume"
                    ],
                    "x-from-ocf": [
                      "volume = ocf.volume"
                    ]
                  }
                },
                "maxvolume": {
                  "type": "integer",
                  "x-ocf-conversion": {
                    "x-ocf-alias": "oic.r.audio",
                    "x-to-ocf": [
                      "range[0] = 0",
                      "range[1] = maxvolume"
                    ],
                    "x-from-ocf": [
                      "maxvolume = range[1]",
                      "otherwise: maxvalue = 100"
                    ]
                  }
                },
                "mute": {
                  "type": "boolean",
                  "x-ocf-conversion": {
                    "x-ocf-alias": "oic.r.audio",
                    "x-to-ocf": [
                      "ocf.mute = mute"
                    ],
```

```
                    "x-from-ocf": [
                      "mute = ocf.mute"
                    ]
                  }
                }
              }
            }
          },
          "type": "object",
          "allOf": [
            {"$ref": "#/definitions/asa.operation.audiovolume"}
          ],
          "required": [ "volume","maxvolume","mute" ]
        }

    responses :
      200:
        body:
          application/json:
            schema: |
              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.operation.audiovolume.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Audio Volume",
                "definitions": {
                  "asa.operation.audiovolume": {
                    "type": "object",
                    "properties": {
                      "volume": {
                        "type": "integer",
                        "description": "Speaker volume index",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.audio",
                          "x-to-ocf": [
                            "ocf.volume = volume"
                          ],
                          "x-from-ocf": [
                            "volume = ocf.volume"
                          ]
                        }
                      },
                      "maxvolume": {
                        "type": "integer",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.audio",
                          "x-to-ocf": [
                            "range[0] = 0",
                            "range[1] = maxvolume"
                          ],
                          "x-from-ocf": [
                            "maxvolume = range[1]",
                            "otherwise: maxvalue = 100"
                          ]
                        }
                      },
                      "mute": {
                        "type": "boolean",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.audio",
                          "x-to-ocf": [
                            "ocf.mute = mute"
                          ],
                          "x-from-ocf": [
                            "mute = ocf.mute"
                          ]
```

```
            }
          }
        }
      }
    },
    "type": "object",
    "allOf": [
      {"$ref": "#/definitions/asa.operation.audiovolume"}
    ],
    "required": [ "volume","maxvolume","mute" ]
  }
```

8.8.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
volume	oic.r.audio	ocf.volume = volume	volume = ocf.volume	Speaker volume index
maxvolume	oic.r.audio	range[0] = 0range[1] = maxvolume	maxvolume = range[1]otherwise: maxvalue = 100	
mute	oic.r.audio	ocf.mute = mute	mute = ocf.mute	

8.8.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/AudioVolumeResURI		get	post		

8.9 Climate Control Mode Mapping

8.9.1 Introduction

This API defines the mapping between an instance of an AllJoyn ClimateControlMode interface and the OCF equivalent Resource. ClimateControlMode has three Properties; these map as follows: mode, supportedmodes -> Mode Resource operationalstate -> OperationalState Resource This can be represented in OCF either as two distinct Resource instances or a single instance with two RTs (oic.r.mode, oic.r.operationalstate)

8.9.2 Example URI

/ClimateControlModeResURI

8.9.3 Resource Type

The resource type (rt) is defined as: oic.r.mode.

8.9.4 RAML Definition

```
#%RAML 0.8

title: ClimateControlModeInterfaceMapping
version: OCFv1.0.0-20170317

traits:
 - interface-actuator :
     queryParameters:
       if:
         enum: ["oic.if.a", "oic.if.baseline"]


/ClimateControlModeResURI:

  description: |
    This API defines the mapping between an instance of an AllJoyn ClimateControlMode interface and
the OCF equivalent Resource.
    ClimateControlMode has three Properties; these map as follows:
      mode, supportedmodes -> Mode Resource
      operationalstate -> OperationalState Resource
    This can be represented in OCF either as two distinct Resource instances or a single instance
```

```
with two RTs (oic.r.mode, oic.r.operationalstate)

  is : ['interface-actuator']
  get:
    responses :
      200:
        body:
          application/json:
            schema: |
              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.operation.climatecontrolmode.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Climate Control Mode",
                "definitions": {
                  "asa.operation.climatecontrolmode": {
                    "type": "object",
                    "properties": {
                      "mode": {
                        "type": "integer",
                        "description": "Current mode of device.",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.mode",
                          "x-to-ocf": [
                            "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                            "ocf.mode[0] = modearray[mode]"
                          ],
                          "x-from-ocf": [
                            "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                            "mode = indexof modeArray[ocf.mode[0]]"
                          ]
                        }
                      },
                      "supportedmodes": {
                        "type": "array",
                        "items": {
                          "type": "integer"
                        },
                        "description": "Array of supported modes",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.mode",
                          "x-to-ocf": [
                            "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                            "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
modearray[supportedmodes[x]]"
                          ],
                          "x-from-ocf": [
                            "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                            "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
modearray[ocf.supportedmodes[x]]"
                          ]
                        }
                      },
                      "operationalstate": {
                        "type": "integer",
                        "description": "Current status of device",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.operationalstate",
                          "x-to-ocf": [
                            "machinestates =
[Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
                            "currentmachinestate = machinestates[operationalstate]"
                          ],
                          "x-from-ocf": [
                            "statearray =
[Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
```

```
                        "operationalstate = indexof statearray[currentmachinestate[0]]"
                      ]
                    }
                  }
                }
              }
            },
            "type": "object",
            "allOf": [
              {"$ref": "#/definitions/asa.operation.climatecontrolmode"}
            ],
            "required": [ "mode","supportedmodes","operationalstate" ]
          }

        example: |
          {
            "rt":     ["oic.r.mode","oic.operationalstate"]
          }

  post:
    body:
      application/json:

        schema: |
          {
            "id":
"http://openinterconnect.org/asamapping/schemas/asa.operation.climatecontrolmode.json#",
            "$schema": "http://json-schema.org/draft-04/schema#",
            "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
            "title": "Climate Control Mode",
            "definitions": {
              "asa.operation.climatecontrolmode": {
                "type": "object",
                "properties": {
                  "mode": {
                    "type": "integer",
                    "description": "Current mode of device.",
                    "x-ocf-conversion": {
                      "x-ocf-alias": "oic.r.mode",
                      "x-to-ocf": [
                        "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                        "ocf.mode[0] = modearray[mode]"
                      ],
                      "x-from-ocf": [
                        "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                        "mode = indexof modeArray[ocf.mode[0]]"
                      ]
                    }
                  },
                  "supportedmodes": {
                    "type": "array",
                    "items": {
                      "type": "integer"
                    },
                    "description": "Array of supported modes",
                    "x-ocf-conversion": {
                      "x-ocf-alias": "oic.r.mode",
                      "x-to-ocf": [
                        "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                        "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
modearray[supportedmodes[x]]"
                      ],
                      "x-from-ocf": [
                        "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                        "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
modearray[ocf.supportedmodes[x]]"
                      ]
                    }
```

```
                },
                "operationalstate": {
                  "type": "integer",
                  "description": "Current status of device",
                  "x-ocf-conversion": {
                    "x-ocf-alias": "oic.r.operationalstate",
                    "x-to-ocf": [
                      "machinestates =
[Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
                      "currentmachinestate = machinestates[operationalstate]"
                    ],
                    "x-from-ocf": [
                      "statearray =
[Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
                      "operationalstate = indexof statearray[currentmachinestate[0]]"
                    ]
                  }
                }
              }
            }
          },
          "type": "object",
          "allOf": [
            {"$ref": "#/definitions/asa.operation.climatecontrolmode"}
          ],
          "required": [ "mode","supportedmodes","operationalstate" ]
        }

    responses :
      200:
        body:
          application/json:
            schema: |
              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.operation.climatecontrolmode.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Climate Control Mode",
                "definitions": {
                  "asa.operation.climatecontrolmode": {
                    "type": "object",
                    "properties": {
                      "mode": {
                        "type": "integer",
                        "description": "Current mode of device.",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.mode",
                          "x-to-ocf": [
                            "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                            "ocf.mode[0] = modearray[mode]"
                          ],
                          "x-from-ocf": [
                            "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                            "mode = indexof modeArray[ocf.mode[0]]"
                          ]
                        }
                      },
                      "supportedmodes": {
                        "type": "array",
                        "items": {
                          "type": "integer"
                        },
                        "description": "Array of supported modes",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.mode",
                          "x-to-ocf": [
```

```
                            "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                            "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
modearray[supportedmodes[x]]"
                          ],
                          "x-from-ocf": [
                            "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                            "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
modearray[ocf.supportedmodes[x]]"
                          ]
                        }
                      },
                      "operationalstate": {
                        "type": "integer",
                        "description": "Current status of device",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.operationalstate",
                          "x-to-ocf": [
                            "machinestates =
[Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
                            "currentmachinestate = machinestates[operationalstate]"
                          ],
                          "x-from-ocf": [
                            "statearray =
[Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
                            "operationalstate = indexof statearray[currentmachinestate[0]]"
                          ]
                        }
                      }
                    }
                  }
                },
                "type": "object",
                "allOf": [
                  {"$ref": "#/definitions/asa.operation.climatecontrolmode"}
                ],
                "required": [ "mode","supportedmodes","operationalstate" ]
              }
```

8.9.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
supportedmodes	oic.r.mode	modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] = modearray[supportedmodes[x]]	modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof modearray[ocf.supportedmodes[x]]	Array of supported modes
operationalstate	oic.r.operationalstate	machinestates = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]currentmachinestate = machinestates[operationalstate]	statearray = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat] operationalstate = indexof statearray[currentmachinestate[0]]	Current status of device
mode	oic.r.mode	modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]ocf.mode[0] = modearray[mode]	modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]mode = indexof modeArray[ocf.mode[0]]	Current mode of device.

8.9.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/ClimateControlModeResURI		get	post		

8.10 Closed Status Mapping

8.10.1 Introduction

This API defines the mapping between an instance of an AllJoyn ClosedStatus Interface and the OCF Door Resource.

8.10.2 Example URI

/ClosedStatusResURI

8.10.3 Resource Type

The resource type (rt) is defined as: oic.r.door.

8.10.4 RAML Definition

```
#%RAML 0.8

title: ClosedStatusInterfaceMapping
version: OCFv1.0.0-20170317

traits:
 - interface-all :
     queryParameters:

       if:
         enum: ["oic.if.s", "oic.if.baseline"]


/ClosedStatusResURI:

  description: |
    This API defines the mapping between an instance of an AllJoyn ClosedStatus Interface and
    the OCF Door Resource.


  is : ['interface-all']

  get:
    responses :

      200:

        body:
          application/json:

            schema: |

              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.operation.closedstatus.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Closed Status",
                "definitions": {
                  "asa.operation.closedstatus": {
                    "type": "object",
                    "properties": {
                      "isclosed": {
                        "type": "boolean",
                        "description": "Open/Closed status Indicator",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.door",
                          "x-to-ocf": [
                            "if isClosed ocf.openState = Closed.",
                            "if !isClosed ocf.openState = Open."
                          ],
                          "x-from-ocf": [
                            "isClosed = (openState == Closed)"
```

```
                ]
              }
            }
          }
        }
      }
    },
    "type": "object",
    "allOf": [
      {"$ref": "#/definitions/asa.operation.closedstatus"}
    ],
    "required": [ "isclosed"]
  }

example: |
  {
    "rt":     ["oic.r.door"]
  }
```

8.10.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
isclosed	oic.r.door	if isClosed ocf.openState = Closed.if !isClosed ocf.openState = Open.	isClosed = (openState == Closed)	Open/Closed status Indicator

8.10.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/ClosedStatusResURI		get			

8.11 Cycle Control Mapping

8.11.1 Introduction

This API defines the mapping between an instance of an AllJoyn CycleControl interface and the OCF OperationalState Resource. The AllJoyn interface also supports a Method, ExecuteOperationalCommand; this is handled in OCF using an instance of oic.r.actuator within an oic.r.action collection. Please see Section 8 of the Mapping Specification for specifics.

8.11.2 Example URI

/CycleControlResURI

8.11.3 Resource Type

The resource type (rt) is defined as: oic.r.operationalstate.

8.11.4 RAML Definition

```
#%RAML 0.8
title: CycleControlInterfaceMapping
version: OCFv1.0.0-20170317
traits:
 - interface-sensor :
     queryParameters:
       if:
         enum: ["oic.if.s", "oic.if.baseline"]


/CycleControlResURI:
  description: |
    This API defines the mapping between an instance of an AllJoyn CycleControl interface and the
OCF OperationalState Resource.
```

```
    The AllJoyn interface also supports a Method, ExecuteOperationalCommand; this is handled in OCF
using an instance of oic.r.actuator within an oic.r.action collection.
    Please see Section 8 of the Mapping Specification for specifics.

  is : ['interface-sensor']
  get:
    responses :
      200:
        body:
          application/json:
            schema: |
              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.operation.ovencyclephase.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Oven Cycle Phase",
                "definitions": {
                  "asa.operation.ovencyclephase": {
                    "type": "object",
                    "properties": {
                      "cyclephase": {
                        "type": "integer",
                        "description": "Current phase of the operational cycle",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.operationalstate",
                          "x-to-ocf": [
                            "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
                            "currentmachinestate = phasearray[cyclephase]"
                          ],
                          "x-from-ocf": [
                            "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
                            "cyclephase = indexof statearray[currentmachinestate[0]]"
                          ]
                        }
                      },
                      "supportedcyclephases": {
                        "type": "array",
                        "items": {
                          "type": "integer"
                        },
                        "description": "Array of cycle phases supported by the Appliance.",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.operationalstate",
                          "x-to-ocf": [
                            "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
                            "for x=0, x < sizeof(supportedcyclephases): machinestates[x] =
phasearray[supportedcyclephases[x]]"
                          ],
                          "x-from-ocf": [
                            "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
                            "for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof
phasearray[machinestates[x]]"
                          ]
                        }
                      },
                      "getvendorphasedescription": {
                        "x-ocf-type": "method",
                        "description": "Get cycle phases description",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.action"
                        }
                      }
                    }
                  }
                },
                "type": "object",
```

```
    "allOf": [
      {"$ref": "#/definitions/asa.operation.ovencyclephase"}
    ],
    "required": [ "cyclephase","supportedcyclephases" ]
  }

example: |
  {
    "rt":     ["oic.r.operationalstate"]
  }
```

8.11.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
supportedcyclephases	oic.r.operationalstate	phasearray = [Unavailable,Preheating,Cooking,Cleaning]for x=0, x < sizeof(supportedcyclephases): machinestates[x] = phasearray[supportedcyclephases[x]]	phasearray = [Unavailable,Preheating,Cooking,Cleaning]for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof phasearray[machinestates[x]]	Array of cyclephases supported by the Appliance.
cyclephase	oic.r.operationalstate	phasearray = [Unavailable,Preheating,Cooking,Cleaning]currentmachinestate = phasearray[cyclephase]	phasearray = [Unavailable,Preheating,Cooking,Cleaning]cyclephase = indexof statearray[currentmachinestate[0]]	Current phase of the operational cycle
getvendorphasedescription	oic.r.action			Get cyclephases description

8.11.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CycleControlResURI		get			

8.12 Fan Speed Level Mapping

8.12.1 Introduction

This API defines the mapping between an instance of an AllJoyn FanSpeedLevel interface and an OCF AirFlow Resource. Note that the setting of the FanSpeedLevel to '0x00' (off) is handled via the 'OffControl' interface rather than writing directly to this interface. In such a case an instance of Binary Switch shall be exposed on the OCF side; this can be modeled as AirFlowControl which is then a collection of Binary Switch and AirFlow.

8.12.2 Example URI

/FanSpeedLevelResURI

8.12.3 Resource Type

The resource type (rt) is defined as: oic.r.airflow.

8.12.4 RAML Definition

```
#%RAML 0.8

title: FanSpeedLevelInterfaceMapping
version: OCFv1.0.0-20170317

traits:
 - interface-actuator :
     queryParameters:

       if:
         enum: ["oic.if.a", "oic.if.baseline"]


/FanSpeedLevelResURI:

  description: |
    This API defines the mapping between an instance of an AllJoyn FanSpeedLevel interface and an
OCF AirFlow Resource.
    Note that the setting of the FanSpeedLevel to '0x00' (off) is handled via the 'OffControl'
interface rather than writing directly to this interface.
    In such a case an instance of Binary Switch shall be exposed on the OCF side; this can be
modeled as AirFlowControl which is then a collection of Binary Switch and AirFlow.


  is : ['interface-actuator']

  get:
    responses :

      200:

        body:
          application/json:

            schema: |

              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.operation.fanspeedlevel.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Fan Speed Level",
                "definitions": {
                  "asa.operation.fanspeedlevel": {
                    "type": "object",
                    "properties": {
                      "fanspeedlevel": {
                        "type": "integer",
                        "description": "Fan speed level. 0 = off.",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.airflow",
                          "x-to-ocf": [
                            "speed = fanspeedlevel"
                          ],
                          "x-from-ocf": [
                            "fanspeedlevel = speed"
                          ]
                        }
                      },
                      "maxfanspeedlevel": {
                        "type": "integer",
                        "description": "Max level allowed for fan speed",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.airflow",
                          "x-to-ocf": [
                            "range[0] = 0",
                            "range[1] = maxfanspeedlevel"
                          ],
                          "x-from-ocf": [
```

```
                        "maxfanspeedlevel = range[1]",
                        "otherwise: maxfanspeedlevel = 100"
                      ]
                    }
                  },
                  "automode": {
                    "type": "integer",
                    "description": "Auto mode status.",
                    "x-ocf-conversion": {
                      "x-ocf-alias": "oic.r.airflow",
                      "x-to-ocf": [
                        "if automode != NotSupported(0xFF)",
                        " ocf.automode = automode",
                        "else no mapping"
                      ],
                      "x-from-ocf": [
                        "automode = ocf.automode",
                        "otherwise: automode = NotSupported(0xFF)"
                      ]
                    }
                  }
                }
              }
            },
            "type": "object",
            "allOf": [
              {"$ref": "#/definitions/asa.operation.fanspeedlevel"}
            ],
            "required": [ "fanspeedlevel","maxfanspeedlevel","automode" ]
          }

        example: |

          {
            "rt":     ["oic.r.airflow"]
          }


  post:
    body:
      application/json:

        schema: |

          {
            "id":
"http://openinterconnect.org/asamapping/schemas/asa.operation.fanspeedlevel.json#",
            "$schema": "http://json-schema.org/draft-04/schema#",
            "description": "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
            "title": "Fan Speed Level",
            "definitions": {
              "asa.operation.fanspeedlevel": {
                "type": "object",
                "properties": {
                  "fanspeedlevel": {
                    "type": "integer",
                    "description": "Fan speed level. 0 = off.",
                    "x-ocf-conversion": {
                      "x-ocf-alias": "oic.r.airflow",
                      "x-to-ocf": [
                        "speed = fanspeedlevel"
                      ],
                      "x-from-ocf": [
                        "fanspeedlevel = speed"
                      ]
                    }
                  },
                  "maxfanspeedlevel": {
                    "type": "integer",
                    "description": "Max level allowed for fan speed",
                    "x-ocf-conversion": {
```

```
                    "x-ocf-alias": "oic.r.airflow",
                    "x-to-ocf": [
                      "range[0] = 0",
                      "range[1] = maxfanspeedlevel"
                    ],
                    "x-from-ocf": [
                      "maxfanspeedlevel = range[1]",
                      "otherwise: maxfanspeedlevel = 100"
                    ]
                  }
                },
                "automode": {
                  "type": "integer",
                  "description": "Auto mode status.",
                  "x-ocf-conversion": {
                    "x-ocf-alias": "oic.r.airflow",
                    "x-to-ocf": [
                      "if automode != NotSupported(0xFF)",
                      " ocf.automode = automode",
                      "else no mapping"
                    ],
                    "x-from-ocf": [
                      "automode = ocf.automode",
                      "otherwise: automode = NotSupported(0xFF)"
                    ]
                  }
                }
              }
            }
          },
          "type": "object",
          "allOf": [
            {"$ref": "#/definitions/asa.operation.fanspeedlevel"}
          ],
          "required": [ "fanspeedlevel","maxfanspeedlevel","automode" ]
        }

    responses :
      200:
        body:
          application/json:
            schema: |
              {
                "id":
"http://openinterconnect.org/samapping/schemas/asa.operation.fanspeedlevel.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Fan Speed Level",
                "definitions": {
                  "asa.operation.fanspeedlevel": {
                    "type": "object",
                    "properties": {
                      "fanspeedlevel": {
                        "type": "integer",
                        "description": "Fan speed level. 0 = off.",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.airflow",
                          "x-to-ocf": [
                            "speed = fanspeedlevel"
                          ],
                          "x-from-ocf": [
                            "fanspeedlevel = speed"
                          ]
                        }
                      },
                      "maxfanspeedlevel": {
                        "type": "integer",
```

```
            "description": "Max level allowed for fan speed",
            "x-ocf-conversion": {
              "x-ocf-alias": "oic.r.airflow",
              "x-to-ocf": [
                "range[0] = 0",
                "range[1] = maxfanspeedlevel"
              ],
              "x-from-ocf": [
                "maxfanspeedlevel = range[1]",
                "otherwise: maxfanspeedlevel = 100"
              ]
            }
          },
          "automode": {
            "type": "integer",
            "description": "Auto mode status.",
            "x-ocf-conversion": {
              "x-ocf-alias": "oic.r.airflow",
              "x-to-ocf": [
                "if automode != NotSupported(0xFF)",
                " ocf.automode = automode",
                "else no mapping"
              ],
              "x-from-ocf": [
                "automode = ocf.automode",
                "otherwise: automode = NotSupported(0xFF)"
              ]
            }
          }
        }
      }
    },
    "type": "object",
    "allOf": [
      {"$ref": "#/definitions/asa.operation.fanspeedlevel"}
    ],
    "required": [ "fanspeedlevel","maxfanspeedlevel","automode" ]
}
```


8.12.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
maxfanspeedlevel	oic.r.airflow	range[0] = 0range[1] = maxfanspeedlevel	maxfanspeedlevel = range[1]otherwise: maxfanspeedlevel = 100	Max level allowed for fan speed
automode	oic.r.airflow	if automode != NotSupported(0xFF) ocf.automode = automodeelse no mapping	automode = ocf.automodeotherwise: automode = NotSupported(0xFF)	Auto mode status.
fanspeedlevel	oic.r.airflow	speed = fanspeedlevel	fanspeedlevel = speed	Fan speed level. 0 = off.

8.12.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/FanSpeedLevelResURI		get	post		

8.13 Heating Zone Mapping

8.13.1 Introduction

This API defines the mapping between an instance of an AllJoyn HeatingZone interface and an OCF HeatingZoneCollection Resource. Each element in the array of heating zones within the

AllJoyn HeatingZone interface maps to an instance of OCF HeatingZone, itself a link in an instance of an OCF HeatingZoneCollection. The mapping defined in the schema describes the population of the OCF HeatingZone Resource that constitutes the Resources that are contained in the collection.

8.13.2 Example URI

/HeatingZoneResURI

8.13.3 Resource Type

The resource type (rt) is defined as: oic.r.heatingzonecollection.

8.13.4 RAML Definition

```
#%RAML 0.8

title: HeatingZoneInterfaceMapping
version: OCFv1.0.0-20170317

traits:
 - interface-sensor :
     queryParameters:

       if:
         enum: ["oic.if.s", "oic.if.baseline"]


/HeatingZoneResURI:

  description: |
    This API defines the mapping between an instance of an AllJoyn HeatingZone interface and an OCF
HeatingZoneCollection Resource.
    Each element in the array of heating zones within the AllJoyn HeatingZone interface maps to an
instance of OCF HeatingZone, itself a link in an instance of an OCF HeatingZoneCollection.
    The mapping defined in the schema describes the population of the OCF HeatingZone Resource that
constitutes the Resources that are contained in the collection.


  is : ['interface-sensor']

  get:
    responses :

      200:

        body:
          application/json:

            schema: |

              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.operation.heatingzone.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Heating Zone",
                "definitions": {
                  "asa.operation.heatingzone": {
                    "type": "object",
                    "properties": {
                      "numberofheatingzones": {
                        "type": "integer",
                        "description": "Number of heating zones.",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.heatingzonecollection",
                          "x-to-ocf": [
                            "number of links in the collection = numberofheatingzones"
                          ],
                          "x-from-ocf": [
                            "numberofheatingzones = number of links in the collection"
                          ]
                        }
                      },
```

```
                    "maxheatinglevels": {
                      "type": "array",
                      "items": {
                        "type": "integer"
                      },
                      "description": "Max heating levels for each zone",
                      "x-ocf-conversion": {
                        "x-ocf-alias": "oic.r.heatingzone",
                        "x-to-ocf": [
                          "Instance of oic.r.heatingzone per array item ",
                          "for x=0, x<sizeof(maxheatinglevels): ocf.maxheatinglevel =
maxheatinglevels[x]"
                        ],
                        "x-from-ocf": [
                          "for x=0;x<numlinks(oic.r.heatingzonecollection): maxheatinglevels[x] =
ocf.maxheatinglevel"
                        ]
                      }
                    },
                    "heatinglevels": {
                      "type": "array",
                      "items": {
                        "type": "integer"
                      },
                      "description": "Current heating levels for each zone.",
                      "x-ocf-conversion": {
                        "x-ocf-alias": "oic.r.heatingzone",
                        "x-to-ocf": [
                          "Instance of oic.r.heatingzone per array item ",
                          "for x=0, x<sizeof(heatinglevels): ocf.heatinglevel =
maxheatinglevels[x]"
                        ],
                        "x-from-ocf": [
                          "for x=0;x<numlinks(oic.r.heatingzonecollection): heatinglevels[x] =
ocf.heatinglevel"
                        ]
                      }
                    }
                  }
                }
              },
              "type": "object",
              "allOf": [
                {"$ref": "#/definitions/asa.operation.heatingzone"}
              ],
              "required": [ "numberofheatingzones","maxheatinglevels","heatinglevels" ]
            }

          example: |
            {
              "rt":     ["oic.r.heatingzonecollection"]
            }
```

8.13.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
heatinglevels	oic.r.heatingzone	Instance of oic.r.heatingzone per array item for x=0, x<sizeof(heatinglevels): ocf.heatinglevel =	for x=0;x<numlinks(oic.r.heatingzonecollection): heatinglevels[x] = ocf.heatinglevel	Current heating levels for each zone.

		maxheatinglevels [x]		
numberofheatingzones	oic.r.heatingzonecollection	number of links in the collection = numberofheatingzones	numberofheatingzones = number of links in the collection	Number of heating zones.
maxheatinglevels	oic.r.heatingzone	Instance of oic.r.heatingzone per array item for x=0, x<sizeof(maxheatinglevels): ocf.maxheatinglevel = maxheatinglevels [x]	for x=0;x<numlinks(oic.r.heatingzonecollection): maxheatinglevels[x] = ocf.maxheatinglevel	Max heating levels for each zone

8.13.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/HeatingZoneResURI		get			

8.14 HVAC Fan Mode Mapping

8.14.1 Introduction

This API defines the mapping between an instance of an AllJoyn HvacFanMode interface and an OCF Mode Resource.

8.14.2 Example URI

/HvacFanModeResURI

8.14.3 Resource Type

The resource type (rt) is defined as: oic.r.mode.

8.14.4 RAML Definition

```
#%RAML 0.8
title: HvacFanModeInterfaceMapping
version: OCFv1.0.0-20170317
traits:
 - interface-actuator :
     queryParameters:
       if:
         enum: ["oic.if.a", "oic.if.baseline"]


/HvacFanModeResURI:
  description: |
    This API defines the mapping between an instance of an AllJoyn HvacFanMode interface and an OCF
Mode Resource.

  is : ['interface-actuator']
  get:
    responses :
      200:
        body:
          application/json:
            schema: |
```

```
              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.operation.hvacfanmode.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "HVAC Fan Mode",
                "definitions": {
                  "asa.operation.hvacfanmode": {
                    "type": "object",
                    "properties": {
                      "mode": {
                        "type": "integer",
                        "description": "Current mode of device.",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.mode",
                          "x-to-ocf": [
                            "modearray = [Auto,Circulation,Continuous]",
                            "ocf.mode[0] = modearray[mode]"
                          ],
                          "x-from-ocf": [
                            "modearray = [Auto,Circulation,Continuous]",
                            "mode = indexof modeArray[ocf.mode[0]]"
                          ]
                        }
                      },
                      "supportedmodes": {
                        "type": "array",
                        "items": {
                          "type": "integer"
                        },
                        "description": "Array of supported modes",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.mode",
                          "x-to-ocf": [
                            "modearray = [Auto,Circulation,Continuous]",
                            "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
modearray[supportedmodes[x]]"
                          ],
                          "x-from-ocf": [
                            "modearray = [Auto,Circulation,Continuous]",
                            "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
modearray[ocf.supportedmodes[x]]"
                          ]
                        }
                      }
                    }
                  }
                },
                "type": "object",
                "allOf": [
                  {"$ref": "#/definitions/asa.operation.hvacfanmode"}
                ],
                "required": [ "mode","supportedmodes" ]
              }


        example: |

              {
                "rt":     ["oic.r.mode"]
              }


  post:
    body:
      application/json:

        schema: |

          {
            "id": "http://openinterconnect.org/asamapping/schemas/asa.operation.hvacfanmode.json#",
            "$schema": "http://json-schema.org/draft-04/schema#",
```

```
            "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
            "title": "HVAC Fan Mode",
            "definitions": {
              "asa.operation.hvacfanmode": {
                "type": "object",
                "properties": {
                  "mode": {
                    "type": "integer",
                    "description": "Current mode of device.",
                    "x-ocf-conversion": {
                      "x-ocf-alias": "oic.r.mode",
                      "x-to-ocf": [
                        "modearray = [Auto,Circulation,Continuous]",
                        "ocf.mode[0] = modearray[mode]"
                      ],
                      "x-from-ocf": [
                        "modearray = [Auto,Circulation,Continuous]",
                        "mode = indexof modeArray[ocf.mode[0]]"
                      ]
                    }
                  },
                  "supportedmodes": {
                    "type": "array",
                    "items": {
                      "type": "integer"
                    },
                    "description": "Array of supported modes",
                    "x-ocf-conversion": {
                      "x-ocf-alias": "oic.r.mode",
                      "x-to-ocf": [
                        "modearray = [Auto,Circulation,Continuous]",
                        "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
modearray[supportedmodes[x]]"
                      ],
                      "x-from-ocf": [
                        "modearray = [Auto,Circulation,Continuous]",
                        "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
modearray[ocf.supportedmodes[x]]"
                      ]
                    }
                  }
                }
              }
            },
            "type": "object",
            "allOf": [
              {"$ref": "#/definitions/asa.operation.hvacfanmode"}
            ],
            "required": [ "mode","supportedmodes" ]
          }

    responses :
      200:
        body:
          application/json:
            schema: |

              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.operation.hvacfanmode.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "HVAC Fan Mode",
                "definitions": {
                  "asa.operation.hvacfanmode": {
                    "type": "object",
                    "properties": {
```

```
                    "mode": {
                      "type": "integer",
                      "description": "Current mode of device.",
                      "x-ocf-conversion": {
                        "x-ocf-alias": "oic.r.mode",
                        "x-to-ocf": [
                          "modearray = [Auto,Circulation,Continuous]",
                          "ocf.mode[0] = modearray[mode]"
                        ],
                        "x-from-ocf": [
                          "modearray = [Auto,Circulation,Continuous]",
                          "mode = indexof modeArray[ocf.mode[0]]"
                        ]
                      }
                    },
                    "supportedmodes": {
                      "type": "array",
                      "items": {
                        "type": "integer"
                      },
                      "description": "Array of supported modes",
                      "x-ocf-conversion": {
                        "x-ocf-alias": "oic.r.mode",
                        "x-to-ocf": [
                          "modearray = [Auto,Circulation,Continuous]",
                          "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
modearray[supportedmodes[x]]"
                        ],
                        "x-from-ocf": [
                          "modearray = [Auto,Circulation,Continuous]",
                          "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
modearray[ocf.supportedmodes[x]]"
                        ]
                      }
                    }
                  }
                },
                "type": "object",
                "allOf": [
                  {"$ref": "#/definitions/esa.operation.hvacfanmode"}
                ],
                "required": [ "mode","supportedmodes" ]
              }
```

8.14.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
supportedmodes	oic.r.mode	modearray = [Auto,Circulation,Continuous]for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] = modearray[supportedmodes[x]]	modearray = [Auto,Circulation,Continuous]for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof modearray[ocf.supportedmodes[x]]	Array of supported modes
mode	oic.r.mode	modearray = [Auto,Circulation,Continuous]ocf.mode[0] = modearray[mode]	modearray = [Auto,Circulation,Continuous]mode = indexof modeArray[ocf.mode[0]]	Current mode of device.

8.14.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify

/HvacFanModeResURI		get	post		

8.15 On Off Mapping

8.15.1 Introduction

This API defines the mapping between an instance of an OCF Binary Switch Resource and the equivalent Interface set by AllJoyn. A discovered instance of a Binary Switch is always mapped to an Operation.OnOffStatus interface. A RETRIEVE on a Binary Switch maps to an action on an instance of an Operation.OnOffStatus Interface. An UPDATE on a Binary Switch maps to a method invocation on either Operation.OnControl or OffControl. value = true maps to Operation.OnControl value = false maps to Operation.OffControl

8.15.2 Example URI

/OnOffResURI

8.15.3 Resource Type

The resource type (rt) is defined as: oic.r.switch.binary.

8.15.4 RAML Definition

```
#%RAML 0.8

title: OnOffInterfaceMapping
version: OCFv1.0.0-20170317

traits:
 - interface-actuator :
     queryParameters:

       if:
         enum: ["oic.if.a", "oic.if.baseline"]

 - interface-all :
     queryParameters:

       if:
         enum: ["oic.if.s", "oic.if.a", "oic.if.baseline"]


/OnOffResURI:

  description: |
    This API defines the mapping between an instance of an OCF Binary Switch Resource and the
    equivalent Interface set by AllJoyn
     A discovered instance of a Binary Switch is always
    mapped to an Operation.OnOffStatus interface.
    A RETRIEVE on a Binary Switch maps to an action on an instance of an Operation.OnOffStatus
Interface.
    An UPDATE on a Binary Switch maps to a method invocation on either Operation.OnControl or
OffControl.
      value = true maps to Operation.OnControl
      value = false maps to Operation.OffControl


  get:

    is : ['interface-all']

    responses :

      200:

        body:
          application/json:

            schema: |

              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.operation.onoffstatus.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "On Off Mapping",
```

```
            "definitions": {
              "asa.operation.onoffstatus": {
                "type": "object",
                "properties": {
                  "onoff": {
                    "type": "boolean",
                    "description": "On/Off status of the device",
                    "x-ocf-conversion": {
                      "x-ocf-alias": "oic.r.switch.binary",
                      "x-to-ocf": [
                        "value = onoff"
                      ],
                      "x-from-ocf": [
                        "onoff = value"
                      ]
                    }
                  }
                }
              }
            },
            "type": "object",
            "allOf": [
              {"$ref": "#/definitions/asa.operation.onoffstatus"}
            ],
            "required": [ "onoff" ]
          }

        example: |
          {
            "rt":     ["oic.r.switch.binary"]
          }

  post:
    is : ['interface-actuator']
    body:
      application/json:
        schema: |
          {
            "id": "http://openinterconnect.org/asamapping/schemas/asa.operation.oncontrol.json#",
            "$schema": "http://json-schema.org/draft-04/schema#",
            "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
            "title": "On/Off Control",
            "definitions": {
              "oic.r.switch.binary": {
                "properties": {
                  "oneOf": [
                    {
                      "properties": {
                        "value": {
                          "type": "boolean",
                          "enum": [true]
                        },
                        "x-ocf-conversion": {
                          "x-x-from-ocf": [
                            "asa.operation.oncontrol::switchon()"
                          ]
                        }
                      }
                    },
                    {
                      "properties": {
                        "value": {
                          "type": "boolean",
                          "enum": [false]
                        },
                        "x-ocf-conversion": {
```

```
                    "x-x-from-ocf": [
                      "asa.operation.offcontrol::switchoff()"
                    ]
                  }
                }
              }
            ]
          }
        },
        "asa.operation.oncontrol": {
          "type": "object",
          "properties": {
            "switchon": {
              "type": "string",
              "format": "method",
              "description": "Turn on the device",
              "x-ocf-conversion": {
                "x-ocf-alias": "oic.r.switch.binary",
                "x-to-ocf": [
                    "value = true"
                ]
              }
            }
          }
        },
        "asa.operation.offcontrol": {
          "type": "object",
          "properties": {
            "switchon": {
              "type": "string",
              "format": "method",
              "description": "Turn off the device",
              "x-ocf-conversion": {
                "x-ocf-alias": "oic.r.switch.binary",
                "x-to-ocf": [
                  "value = false"
                ]
              }
            }
          }
        }
      },
      "type": "object",
      "oneOf": [
        {"$ref": "#/definitions/oic.r.switch.binary"},
        {"$ref": "#/definitions/asa.operation.oncontrol"},
        {"$ref": "#/definitions/asa.operation.offcontrol"}
      ]
    }

  responses :
    200:
      body:
        application/json:
          schema: |
            {
              "id":
"http://openinterconnect.org/asamapping/schemas/asa.operation.oncontrol.json#",
              "$schema": "http://json-schema.org/draft-04/schema#",
              "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
              "title": "On/Off Control",
              "definitions": {
                "oic.r.switch.binary": {
                  "properties": {
                    "oneOf": [
                      {
                        "properties": {
```

```
                "value": {
                  "type": "boolean",
                  "enum": [true]
                },
                "x-ocf-conversion": {
                  "x-x-from-ocf": [
                    "asa.operation.oncontrol::switchon()"
                  ]
                }
              }
            },
            {
              "properties": {
                "value": {
                  "type": "boolean",
                  "enum": [false]
                },
                "x-ocf-conversion": {
                  "x-x-from-ocf": [
                    "asa.operation.offcontrol::switchoff()"
                  ]
                }
              }
            }
          ]
        }
      },
      "asa.operation.oncontrol": {
        "type": "object",
        "properties": {
          "switchon": {
            "type": "string",
            "format": "method",
            "description": "Turn on the device",
            "x-ocf-conversion": {
              "x-ocf-alias": "oic.r.switch.binary",
              "x-to-ocf": [
                  "value = true"
              ]
            }
          }
        }
      },
      "asa.operation.offcontrol": {
        "type": "object",
        "properties": {
          "switchon": {
            "type": "string",
            "format": "method",
            "description": "Turn off the device",
            "x-ocf-conversion": {
              "x-ocf-alias": "oic.r.switch.binary",
              "x-to-ocf": [
                "value = false"
              ]
            }
          }
        }
      }
    },
    "type": "object",
    "oneOf": [
      {"$ref": "#/definitions/oic.r.switch.binary"},
      {"$ref": "#/definitions/asa.operation.oncontrol"},
      {"$ref": "#/definitions/asa.operation.offcontrol"}
    ]
  }
```

8.15.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
onoff	oic.r.switch.binary	value = onoff	onoff = value	On/Off status of the device

8.15.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/OnOffResURI		get	post		

8.16 Oven Cycle Phase Mapping

8.16.1 Introduction

This API defines the mapping between an instance of an AllJoyn OvenCyclePhase interface and the OCF OperationalState Resource. OvenCyclePhase cyclephase Property pre-defines values 0x00-0x7F, 0x80-0xFF is for vendor specific values. The mapping defined herein covers only Spec defined values. Any vendor defined value shall be represented in OCF using the x.<organization> syntax for a vendor defined Property. The AllJoyn interface also supports a Method, GetVendorPhasesDescription; this is handled in OCF using an instance of oic.r.actuator within an oic.r.action collection. Please see Section 8 of the Mapping Specification for specifics.

8.16.2 Example URI

/OvenCyclePhaseResURI

8.16.3 Resource Type

The resource type (rt) is defined as: oic.r.operationalstate.

8.16.4 RAML Definition

```
#%RAML 0.8

title: OvenCyclePhaseInterfaceMapping
version: OCFv1.0.0-20170317

traits:
 - interface-sensor :
     queryParameters:

       if:
         enum: ["oic.if.s", "oic.if.baseline"]


/OvenCyclePhaseResURI:

  description: |
    This API defines the mapping between an instance of an AllJoyn OvenCyclePhase interface and the
OCF OperationalState Resource.
    OvenCyclePhase cyclephase Property pre-defines values 0x00-0x7F, 0x80-0xFF is for vendor
specific values.
    The mapping defined herein covers only Spec defined values.
    Any vendor defined value shall be represented in OCF using the x.<organization> syntax for a
vendor defined Property.
    The AllJoyn interface also supports a Method, GetVendorPhasesDescription; this is handled in
OCF using an instance of oic.r.actuator within an oic.r.action collection.
    Please see Section 8 of the Mapping Specification for specifics.


  is : ['interface-sensor']

  get:
    responses :

      200:

        body:
          application/json:

            schema: |
```

```
              {
                "id":
"http://openinterconnect.org/asamapping/schemas/asa.operation.ovencyclephase.json#",
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "title": "Oven Cycle Phase",
                "definitions": {
                  "asa.operation.ovencyclephase": {
                    "type": "object",
                    "properties": {
                      "cyclephase": {
                        "type": "integer",
                        "description": "Current phase of the operational cycle",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.operationalstate",
                          "x-to-ocf": [
                            "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
                            "currentmachinestate = phasearray[cyclephase]"
                          ],
                          "x-from-ocf": [
                            "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
                            "cyclephase = indexof statearray[currentmachinestate[0]]"
                          ]
                        }
                      },
                      "supportedcyclephases": {
                        "type": "array",
                        "items": {
                          "type": "integer"
                        },
                        "description": "Array of cycle phases supported by the Appliance.",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.operationalstate",
                          "x-to-ocf": [
                            "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
                            "for x=0, x < sizeof(supportedcyclephases): machinestates[x] =
phasearray[supportedcyclephases[x]]"
                          ],
                          "x-from-ocf": [
                            "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
                            "for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof
phasearray[machinestates[x]]"
                          ]
                        }
                      },
                      "getvendorphasedescription": {
                        "x-ocf-type": "method",
                        "description": "Get cycle phases description",
                        "x-ocf-conversion": {
                          "x-ocf-alias": "oic.r.action"
                        }
                      }
                    }
                  }
                },
                "type": "object",
                "allOf": [
                  {"$ref": "#/definitions/asa.operation.ovencyclephase"}
                ],
                "required": [ "cyclephase","supportedcyclephases" ]
              }

            example: |
              {
                "rt":     ["oic.r.operationalstate"]
              }
```

8.16.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
supportedcyclephases	oic.r.operationalstate	phasearray = [Unavailable,Preheating,Cooking,Cleaning]for x=0, x < sizeof(supportedcyclephases): machinestates[x] = phasearray[supportedcyclephases[x]]	phasearray = [Unavailable,Preheating, Cooking,Cleaning]for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof phasearray[machinestates[x]]	Array of cycle phases supported by the Appliance.
cyclephase	oic.r.operationalstate	phasearray = [Unavailable,Preheating,Cooking,Cleaning]currentmachinestate = phasearray[cyclephase]	phasearray = [Unavailable,Preheating, Cooking,Cleaning]cyclephase = indexof statearray[currentmachinestate[0]]	Current phase of the operational cycle
getvendorphasedescription	oic.r.action			Get cycle phases description

8.16.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/OvenCyclePhaseResURI		get			

Annex A Swagger2.0 (Informative)

A.1 Audio Volume Mapping

A.1.1 Introduction

This API defines the mapping between an instance of an OCF Audio Controls and the AllJoyn Audio Volume interface.

A.1.2 Example URI

/AudioVolumeResURI

A.1.3 Resource Type

The resource type (rt) is defined as: ['oic.r.audio'].

A.1.4 Swagger2.0 Definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Audio Volume Mapping",
    "version": "OCFv1.0.0-20170317",
    "license": {
      "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n        1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2.  Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/AudioVolumeResURI" : {
      "get": {
        "description": "This API defines the mapping between an instance of an OCF Audio
Controls\n and the AllJoyn Audio Volume interface.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-all"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt":     ["oic.r.audio"]
                }
                ,
              "schema": { "$ref": "#/definitions/RetrieveSchema" }
            }
        }
      },
```

```
    "post": {
      "description": "",
      "parameters": [
        {"$ref": "#/parameters/interface-all"},
        {
          "name": "body",
          "in": "body",
          "required": true,
          "schema": { "$ref": "#/definitions/UpdateSchema" }
        }
      ],
      "responses": {
          "200": {
            "description" : "",
            "schema": { "$ref": "#/definitions/UpdateSchema" }
          }
      }
    }
  }
},
"parameters": {
  "interface-all" : {
    "in" : "query",
    "name" : "if",
    "type" : "string",
    "enum" : ["oic.if.a", "oic.if.baseline"]
  }
},
"definitions": {
  "RetrieveSchema" :
          {
      "properties": {
        "maxvolume": {
          "type": "integer",
          "x-ocf-conversion": {
            "x-from-ocf": [
              "maxvolume = range[1]",
              "otherwise: maxvalue = 100"
            ],
            "x-ocf-alias": "oic.r.audio",
            "x-to-ocf": [
              "range[0] = 0",
              "range[1] = maxvolume"
            ]
          }
        },
        "mute": {
          "type": "boolean",
          "x-ocf-conversion": {
            "x-from-ocf": [
              "mute = ocf.mute"
            ],
            "x-ocf-alias": "oic.r.audio",
            "x-to-ocf": [
              "ocf.mute = mute"
            ]
          }
        },
        "volume": {
          "description": "Speaker volume index",
          "type": "integer",
          "x-ocf-conversion": {
            "x-from-ocf": [
              "volume = ocf.volume"
            ],
            "x-ocf-alias": "oic.r.audio",
            "x-to-ocf": [
              "ocf.volume = volume"
            ]
          }
        }
```

```
        },
        "required": [
          "volume",
          "maxvolume",
          "mute"
        ],
        "type": "object"
      }

    ,
    "UpdateSchema" :
          {
        "properties": {
          "maxvolume": {
            "type": "integer",
            "x-ocf-conversion": {
              "x-from-ocf": [
                "maxvolume = range[1]",
                "otherwise: maxvalue = 100"
              ],
              "x-ocf-alias": "oic.r.audio",
              "x-to-ocf": [
                "range[0] = 0",
                "range[1] = maxvolume"
              ]
            }
          },
          "mute": {
            "type": "boolean",
            "x-ocf-conversion": {
              "x-from-ocf": [
                "mute = ocf.mute"
              ],
              "x-ocf-alias": "oic.r.audio",
              "x-to-ocf": [
                "ocf.mute = mute"
              ]
            }
          },
          "volume": {
            "description": "Speaker volume index",
            "type": "integer",
            "x-ocf-conversion": {
              "x-from-ocf": [
                "volume = ocf.volume"
              ],
              "x-ocf-alias": "oic.r.audio",
              "x-to-ocf": [
                "ocf.volume = volume"
              ]
            }
          }
        },
        "required": [
          "volume",
          "maxvolume",
          "mute"
        ],
        "type": "object"
      }

  }
}
```

A.1.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description

volume	oic.r.audio	ocf.volume = volume	volume = ocf.volume	Speaker volume index
maxvolume	oic.r.audio	range[0] = 0range[1] = maxvolume	maxvolume = range[1]otherwise: maxvalue = 100	
mute	oic.r.audio	ocf.mute = mute	mute = ocf.mute	

A.1.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/AudioVolumeResURI		get	post		

A.2 Climate Control Mode Mapping

A.2.1 Introduction

This API defines the mapping between an instance of an AllJoyn ClimateControlMode interface and the OCF equivalent Resource. ClimateControlMode has three Properties; these map as follows: mode, supportedmodes -> Mode Resource operationalstate -> OperationalState Resource This can be represented in OCF either as two distinct Resource instances or a single instance with two RTs (oic.r.mode, oic.r.operationalstate)

A.2.2 Example URI

/ClimateControlModeResURI

A.2.3 Resource Type

The resource type (rt) is defined as: ['oic.r.mode', 'oic.operationalstate'].

A.2.4 Swagger2.0 Definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Climate Control Mode Mapping",
    "version": "OCFv1.0.0-20170317",
    "license": {
      "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n        1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2.  Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/ClimateControlModeResURI" : {
      "get": {
        "description": "This API defines the mapping between an instance of an AllJoyn
ClimateControlMode interface and the OCF equivalent Resource.\nClimateControlMode has three
Properties; these map as follows:\n  mode, supportedmodes -> Mode Resource\n  operationalstate ->
```

```
OperationalState Resource\nThis can be represented in OCF either as two distinct Resource instances
or a single instance with two RTs (oic.r.mode, oic.r.operationalstate)\n",
        "parameters": [
          {"$ref": "#/parameters/interface-actuator"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt":     ["oic.r.mode","oic.operationalstate"]
                }
                ,
              "schema": { "$ref": "#/definitions/RetrieveSchema" }
            }
        }
      },
      "post": {
        "description": "",
        "parameters": [
          {"$ref": "#/parameters/interface-actuator"},
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": { "$ref": "#/definitions/UpdateSchema" }
          }
        ],
        "responses": {
            "200": {
              "description" : "",
              "schema": { "$ref": "#/definitions/UpdateSchema" }
            }
        }
      }
    }
  },
  "parameters": {
    "interface-actuator" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.a", "oic.if.baseline"]
    }
  },
  "definitions": {
    "RetrieveSchema" :
            {
        "properties": {
          "mode": {
            "description": "Current mode of device.",
            "type": "integer",
            "x-ocf-conversion": {
              "x-from-ocf": [
                "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                "mode = indexof modeArray[ocf.mode[0]]"
              ],
              "x-ocf-alias": "oic.r.mode",
              "x-to-ocf": [
                "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                "ocf.mode[0] = modearray[mode]"
              ]
            }
          },
          "operationalstate": {
            "description": "Current status of device",
            "type": "integer",
            "x-ocf-conversion": {
              "x-from-ocf": [
                "statearray = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
                "operationalstate = indexof statearray[currentmachinestate[0]]"
```

```
              ],
              "x-ocf-alias": "oic.r.operationalstate",
              "x-to-ocf": [
                "machinestates = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
                "currentmachinestate = machinestates[operationalstate]"
              ]
            }
          },
          "supportedmodes": {
            "description": "Array of supported modes",
            "items": {
              "type": "integer"
            },
            "type": "array",
            "x-ocf-conversion": {
              "x-from-ocf": [
                "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
modearray[ocf.supportedmodes[x]]"
              ],
              "x-ocf-alias": "oic.r.mode",
              "x-to-ocf": [
                "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
modearray[supportedmodes[x]]"
              ]
            }
          }
        },
        "required": [
          "mode",
          "supportedmodes",
          "operationalstate"
        ],
        "type": "object"
      }

    ,
    "UpdateSchema" :
            {
        "properties": {
          "mode": {
            "description": "Current mode of device.",
            "type": "integer",
            "x-ocf-conversion": {
              "x-from-ocf": [
                "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                "mode = indexof modeArray[ocf.mode[0]]"
              ],
              "x-ocf-alias": "oic.r.mode",
              "x-to-ocf": [
                "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                "ocf.mode[0] = modearray[mode]"
              ]
            }
          },
          "operationalstate": {
            "description": "Current status of device",
            "type": "integer",
            "x-ocf-conversion": {
              "x-from-ocf": [
                "statearray = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
                "operationalstate = indexof statearray[currentmachinestate[0]]"
              ],
              "x-ocf-alias": "oic.r.operationalstate",
              "x-to-ocf": [
                "machinestates = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
                "currentmachinestate = machinestates[operationalstate]"
              ]
            }
          },
```

```
          "supportedmodes": {
            "description": "Array of supported modes",
            "items": {
              "type": "integer"
            },
            "type": "array",
            "x-ocf-conversion": {
              "x-from-ocf": [
                "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
modearray[ocf.supportedmodes[x]]"
              ],
              "x-ocf-alias": "oic.r.mode",
              "x-to-ocf": [
                "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
                "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
modearray[supportedmodes[x]]"
              ]
            }
          }
        },
        "required": [
          "mode",
          "supportedmodes",
          "operationalstate"
        ],
        "type": "object"
      }

  }
}
```

A.2.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
supportedmodes	oic.r.mode	modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] = modearray[supportedmodes[x]]	modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof modearray[ocf.supportedmodes[x]]	Array of supported modes
operationalstate	oic.r.operationalstate	machinestates = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]currentmachinestate = machinestates[operationalstate]	statearray = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat] operationalstate = indexof statearray[currentmachinestate[0]]	Current status of device
mode	oic.r.mode	modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]ocf.mode[0] = modearray[mode]	modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]mode = indexof modeArray[ocf.mode[0]]	Current mode of device.

A.2.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/ClimateControlModeResURI		get	post		

A.3 Closed Status Mapping

A.3.1 Introduction

This API defines the mapping between an instance of an AllJoyn ClosedStatus Interface and the OCF Door Resource.

A.3.2 Example URI

/ClosedStatusResURI

A.3.3 Resource Type

The resource type (rt) is defined as: ['oic.r.door'].

A.3.4 Swagger2.0 Definition


```
{
  "swagger": "2.0",
  "info": {
    "title": "Closed Status Mapping",
    "version": "OCFv1.0.0-20170317",
    "license": {
      "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n        1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2.  Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/ClosedStatusResURI" : {
      "get": {
        "description": "This API defines the mapping between an instance of an AllJoyn ClosedStatus
Interface and\nthe OCF Door Resource.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-all"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt":     ["oic.r.door"]
                }
                ,
              "schema": { "$ref": "#/definitions/RetrieveSchema" }
            }
        }
      }
    }
  },
  "parameters": {
    "interface-all" : {
      "in" : "query",
      "name" : "if",
```

```
      "type" : "string",
      "enum" : ["oic.if.s", "oic.if.baseline"]
    }
  },
  "definitions": {
    "RetrieveSchema" :
          {
        "properties": {
          "isclosed": {
            "description": "Open/Closed status Indicator",
            "type": "boolean",
            "x-ocf-conversion": {
              "x-from-ocf": [
                "isClosed = (openState == Closed)"
              ],
              "x-ocf-alias": "oic.r.door",
              "x-to-ocf": [
                "if isClosed ocf.openState = Closed.",
                "if !isClosed ocf.openState = Open."
              ]
            }
          }
        },
        "required": [
          "isclosed"
        ],
        "type": "object"
      }

  }
}
```

A.3.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
isclosed	oic.r.door	if isClosed ocf.openState = Closed.if !isClosed ocf.openState = Open.	isClosed = (openState == Closed)	Open/Closed status Indicator

A.3.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/ClosedStatusResURI		get			

A.4 Air Quality Mapping

A.4.1 Introduction

This API defines the mapping between the AllJoyn AirQuality interface and the OCF AirQuality Resource.
If more than one instance of the AirQuality interface is exposed then each instance maps to an instance of the OCF AirQuality Resource.
The mapping defined in the schema describes the population of the OCF AirQuality Resource.
Even if there is only a single instance of an OCF AirQuality Resource this shall be included in an instance of an OCF AirQualityCollection.
The number of links in the collection equates to the number of instances of the AllJoyn CurrentAirQuality interface that are exposed.
When mapping from OCF the valueType of the Resource shall be introspected, this API is invoked only if this is set to 'Measured'

A.4.2 Example URI

/CurrentAirQualityResURI

A.4.3 Resource Type

The resource type (rt) is defined as: ['oic.r.airqualitycollection'].

A.4.4 Swagger2.0 Definition


```
{
  "swagger": "2.0",
  "info": {
    "title": "Air Quality Mapping",
    "version": "OCFv1.0.0-20170317",
    "license": {
      "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n        1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2.  Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/CurrentAirQualityResURI" : {
      "get": {
        "description": "This API defines the mapping between the AllJoyn AirQuality interface and
the OCF AirQuality Resource.\nIf more than one instance of the AirQuality interface is exposed then
each instance maps to an instance of the OCF AirQuality Resource.\nThe mapping defined in the
schema describes the population of the OCF AirQuality Resource.\nEven if there is only a single
instance of an OCF AirQuality Resource this shall be included in an instance of an OCF
AirQualityCollection.\nThe number of links in the collection equates to the number of instances of
the AllJoyn CurrentAirQuality interface that are exposed.\nWhen mapping from OCF the valueType of
the Resource shall be introspected, this API is invoked only if this is set to 'Measured'\n",
        "parameters": [
          {"$ref": "#/parameters/interface-sensor"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt":     ["oic.r.airqualitycollection"]
                }
                ,
              "schema": { "$ref": "#/definitions/RetrieveSchema" }
            }
        }
      }
    }
  },
  "parameters": {
    "interface-sensor" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.s", "oic.if.baseline"]
    }
```

```
},
"definitions": {
  "RetrieveSchema" :
        {
      "properties": {
        "contaminanttype": {
          "description": "The contaminant type",
          "type": "integer",
          "x-ocf-conversion": {
            "x-from-ocf": [
              "contaminanttype = indexof contaminanttypearray[ocf.contaminanttype]"
            ],
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "valuetype = Measured",
              "contaminanttypearray = [CH2O,CO2,CO,PM2_5,PM10,VOC]",
              "ocf.contaminanttype = contaminanttypearray[contaminanttype]"
            ]
          }
        },
        "currentvalue": {
          "type": "number",
          "x-ocf-conversion": {
            "x-from-ocf": [
              "currentvalue = contaminantvalue"
            ],
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "contaminantvalue = currentvalue"
            ]
          }
        },
        "maxvalue": {
          "type": "number",
          "x-ocf-conversion": {
            "x-from-ocf": [
              "maxvalue = range[1]"
            ],
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "range[1] = maxvalue"
            ]
          }
        },
        "minvalue": {
          "type": "number",
          "x-ocf-conversion": {
            "x-from-ocf": [
              "minvalue = range[0]"
            ],
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "range[0] = minvalue"
            ]
          }
        },
        "precision": {
          "type": "number",
          "x-ocf-conversion": {
            "x-from-ocf": [
              "precision = ocf.precision"
            ],
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "ocf.precision = precision"
            ]
          }
        },
        "updatemintime": {
          "type": "integer",
          "x-ocf-conversion": {
```

```
            "x-from-ocf": [
              "updatemintime = ocf.minnotifyperiod"
            ],
            "x-ocf-alias": "oic.r.value.conditional",
            "x-to-ocf": [
              "ocf.minnotifyperiod = updatemintime"
            ]
          }
        }
      },
      "required": [
        "contaminanttype",
        "currentvalue",
        "minvalue",
        "maxvalue",
        "precision",
        "updatemintime"
      ],
      "type": "object"
    }

  }
}
```

A.4.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
currentvalue	oic.r.airquality	contaminantvalue = currentvalue	currentvalue = contaminantvalue	
updatemintime	oic.r.value.conditional	ocf.minnotifyperiod = updatemintime	updatemintime = ocf.minnotifyperiod	
maxvalue	oic.r.airquality	range[1] = maxvalue	maxvalue = range[1]	
precision	oic.r.airquality	ocf.precision = precision	precision = ocf.precision	
minvalue	oic.r.airquality	range[0] = minvalue	minvalue = range[0]	
contaminanttype	oic.r.airquality	valuetype = Measuredcontaminanttypearray = [CH2O,CO2,CO,PM2_5,PM10,VOC]ocf.contaminanttype = contaminanttypearray[contaminanttype]	contaminanttype = indexof contaminanttypearray[ocf.contaminanttype]	The contaminant type

A.4.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentAirQualityResURI		get			

A.5 Air Quality Level Mapping

A.5.1 Introduction

This API defines the mapping between the AllJoyn AirQualityLevel interface and the OCF AirQuality Resource.
If more than one instance of the AirQualityLevel interface is exposed then each instance maps to an instance of the OCF AirQuality Resource.
The mapping defined in the schema describes the population of the OCF AirQuality Resource.

Even if there is only a single instance of an OCF AirQuality Resource then this shall be included in an instance of an OCF AirQualityCollection.
The number of links in the collection equates to the number of instances of the AllJoyn CurrentAirQuality interface that are exposed.
When mapping from OCF the valueType of the Resource shall be introspected, this API is invoked only if this is set to 'Qualitative'

A.5.2 Example URI

/CurrentAirQualityLevelResURI

A.5.3 Resource Type

The resource type (rt) is defined as: ['oic.r.airqualitycollection'].

A.5.4 Swagger2.0 Definition


```
{
  "swagger": "2.0",
  "info": {
    "title": "Air Quality Level Mapping",
    "version": "OCFv1.0.0-20170317",
    "license": {
      "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n        1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2.  Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/CurrentAirQualityLevelResURI" : {
      "get": {
        "description": "This API defines the mapping between the AllJoyn AirQualityLevel interface
and the OCF AirQuality Resource.\nIf more than one instance of the AirQualityLevel interface is
exposed then each instance maps to an instance of the OCF AirQuality Resource.\nThe mapping defined
in the schema describes the population of the OCF AirQuality Resource.\nEven if there is only a
single instance of an OCF AirQuality Resource then this shall be included in an instance of an OCF
AirQualityCollection.\nThe number of links in the collection equates to the number of instances of
the AllJoyn CurrentAirQuality interface that are exposed.\nWhen mapping from OCF the valueType of
the Resource shall be introspected, this API is invoked only if this is set to 'Qualitative'\n",
        "parameters": [
          {"$ref": "#/parameters/interface-sensor"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt":     ["oic.r.airqualitycollection"]
                }
                ,
              "schema": { "$ref": "#/definitions/RetrieveSchema" }
            }
        }
```

```
    }
  }
},
"parameters": {
  "interface-sensor" : {
    "in" : "query",
    "name" : "if",
    "type" : "string",
    "enum" : ["oic.if.s", "oic.if.baseline"]
  }
},
"definitions": {
  "RetrieveSchema" :
          {
      "properties": {
        "contaminanttype": {
          "description": "The contaminant type",
          "type": "integer",
          "x-ocf-conversion": {
            "x-from-ocf": [
              "if ocf.contaminanttype = CH2O, contaminanttype = 0",
              "if ocf.contaminanttype = CO2, contaminanttype = 1",
              "if ocf.contaminanttype = CO, contaminanttype = 2",
              "if ocf.contaminanttype = PM2_5, contaminanttype = 3",
              "if ocf.contaminanttype = PM10, contaminanttype = 4",
              "if ocf.contaminanttype = VOC, contaminanttype = 5",
              "if ocf.contaminanttype = Smoke, contaminanttype = 253",
              "if ocf.contaminanttype = Odor, contaminanttype = 254",
              "if ocf.contaminanttype = AirPollution, contaminanttype = 255"
            ],
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "valuetype = Qualitative",
              "if contaminanttype = 0, ocf.contaminanttype = CH2O",
              "if contaminanttype = 1, ocf.contaminanttype = CO2",
              "if contaminanttype = 2, ocf.contaminanttype = CO",
              "if contaminanttype = 3, ocf.contaminanttype = PM2_5",
              "if contaminanttype = 4, ocf.contaminanttype = PM10",
              "if contaminanttype = 5, ocf.contaminanttype = VOC",
              "if contaminanttype = 253, ocf.contaminanttype = Smoke",
              "if contaminanttype = 254, ocf.contaminanttype = Odor",
              "if contaminanttype = 255, ocf.contaminanttype = AirPollution"
            ]
          }
        },
        "currentlevel": {
          "type": "integer",
          "x-ocf-conversion": {
            "x-from-ocf": [
              "currentlevel = contaminantvalue"
            ],
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "contaminantvalue = currentlevel"
            ]
          }
        },
        "maxlevel": {
          "type": "integer",
          "x-ocf-conversion": {
            "x-from-ocf": [
              "maxvalue = range[1]"
            ],
            "x-ocf-alias": "oic.r.airquality",
            "x-to-ocf": [
              "range[0] = 0",
              "range[1] = maxvalue"
            ]
          }
        }
      },
```

```
        "required": [
          "contaminanttype",
          "currentlevel",
          "maxlevel"
        ],
        "type": "object"
      }

  }
}
```

A.5.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
currentlevel	oic.r.airquality	contaminantvalue = currentlevel	currentlevel = contaminantvalue	
maxlevel	oic.r.airquality	range[0] = 0range[1] = maxvalue	maxvalue = range[1]	
contaminanttype	oic.r.airquality	valuetype = Qualitativeif contaminanttype = 0, ocf.contaminanttype = CH2Oif contaminanttype = 1, ocf.contaminanttype = CO2if contaminanttype = 2, ocf.contaminanttype = COif contaminanttype = 3, ocf.contaminanttype = PM2_5if contaminanttype = 4, ocf.contaminanttype = PM10if contaminanttype = 5, ocf.contaminanttype = VOCif contaminanttype = 253, ocf.contaminanttype = Smokeif contaminanttype = 254, ocf.contaminanttype = Odorif contaminanttype = 255, ocf.contaminanttype = AirPollution	if ocf.contaminanttype = CH2O, contaminanttype = 0if ocf.contaminanttype = CO2, contaminanttype = 1if ocf.contaminanttype = CO, contaminanttype = 2if ocf.contaminanttype = PM2_5, contaminanttype = 3if ocf.contaminanttype = PM10, contaminanttype = 4if ocf.contaminanttype = VOC, contaminanttype = 5if ocf.contaminanttype = Smoke, contaminanttype = 253if ocf.contaminanttype = Odor, contaminanttype = 254if ocf.contaminanttype = AirPollution, contaminanttype = 255	The contaminant type

A.5.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentAirQualityLevelResURI		get			

A.6 Current Humidity Mapping

A.6.1 Introduction

This API defines the mapping between an instance of an OCF Humidity which exposes only a sensor interface and the AllJoyn Current Humidity interface. A RETRIEVE on a Temperature Sensor maps to an action on an instance of an Environment.CurrentTemperature Interface.

A.6.2 Example URI

/CurrentHumidityResURI

A.6.3 Resource Type

The resource type (rt) is defined as: ['oic.r.humidity'].

A.6.4 Swagger2.0 Definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Current Humidity Mapping",
    "version": "OCFv1.0.0-20170317",
    "license": {
      "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n        1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2.  Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/CurrentHumidityResURI" : {
      "get": {
        "description": "This API defines the mapping between an instance of an OCF Humidity which
exposes only a sensor interface\nand the AllJoyn Current Humidity interface.\nA RETRIEVE on a
Temperature Sensor maps to an action on an instance of an Environment.CurrentTemperature
Interface.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-sensor"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt":     ["oic.r.humidity"]
                }
```

```
            ,
            "schema": { "$ref": "#/definitions/RetrieveSchema" }
          }
        }
      }
    }
  },
  "parameters": {
    "interface-sensor" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.s", "oic.if.baseline"]
    }
  },
  "definitions": {
    "RetrieveSchema" :
          {
        "properties": {
          "currentvalue": {
            "description": "Measured value",
            "type": "number",
            "x-ocf-conversion": {
              "x-from-ocf": [
                "currentvalue = humidity"
              ],
              "x-ocf-alias": "oic.r.humidity",
              "x-to-ocf": [
                "humidity = currentValue"
              ]
            }
          },
          "maxvalue": {
            "description": "Max measured value for humidty",
            "type": "number",
            "x-ocf-conversion": {
              "x-from-ocf": [
                "maxvalue = range[1]"
              ],
              "x-ocf-alias": "oic.r.humidity",
              "x-to-ocf": [
                "range[0] = 0",
                "range[1] = maxvalue"
              ]
            }
          }
        },
        "required": [
          "currentvalue",
          "maxvalue"
        ],
        "type": "object"
      }

  }
}
```


A.6.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
currentvalue	oic.r.humidity	humidity = currentValue	currentvalue = humidity	Measured value
maxvalue	oic.r.humidity	range[0] = 0range[1] = maxvalue	maxvalue = range[1]	Max measured value for humidty

A.6.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentHumidityResURI		get			

A.7 Current Temperature Mapping

A.7.1 Introduction

This API defines the mapping between an instance of an OCF Temperature which exposes only a sensor interface and the AllJoyn Current Temperature interface.
A RETRIEVE on a Temperature Sensor maps to an action on an instance of an Environment.CurrentTemperature Interface.

A.7.2 Example URI

/CurrentTemperatureResURI

A.7.3 Resource Type

The resource type (rt) is defined as: ['oic.r.temperature'].

A.7.4 Swagger2.0 Definition


```
{
  "swagger": "2.0",
  "info": {
    "title": "Current Temperature Mapping",
    "version": "OCFv1.0.0-20170317",
    "license": {
      "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n        1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2.  Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/CurrentTemperatureResURI" : {
      "get": {
        "description": "This API defines the mapping between an instance of an OCF Temperature
which exposes only a sensor interface\nand the AllJoyn Current Temperature interface.\nA RETRIEVE
on a Temperature Sensor maps to an action on an instance of an Environment.CurrentTemperature
Interface.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-sensor"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt":     ["oic.r.temperature"]
                }
```

```
            ,
            "schema": { "$ref": "#/definitions/RetrieveSchema" }
          }
        }
      }
    }
  },
  "parameters": {
    "interface-sensor" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.s", "oic.if.baseline"]
    }
  },
  "definitions": {
    "RetrieveSchema" :
          {
        "properties": {
          "currentvalue": {
            "description": "Measured value",
            "type": "number",
            "x-ocf-conversion": {
              "x-from-ocf": {
                "oneOf": [
                  {
                    "properties": {
                      "enum": [
                        "C"
                      ],
                      "units": "string"
                    },
                    "x-from-ocf": [
                      "currentvalue = temperature"
                    ]
                  },
                  {
                    "properties": {
                      "enum": [
                        "F"
                      ],
                      "units": "string"
                    },
                    "x-from-ocf": [
                      "currentvalue = (temperature-32)*5/9"
                    ]
                  },
                  {
                    "properties": {
                      "enum": [
                        "K"
                      ],
                      "units": "string"
                    },
                    "x-from-ocf": [
                      "currentvalue = temperature-273.15"
                    ]
                  }
                ]
              },
              "x-ocf-alias": "oic.r.temperature",
              "x-to-ocf": [
                "temperature = currentValue",
                "units = C"
              ]
            }
          },
          "precision": {
            "type": "number",
            "x-ocf-conversion": {
              "x-from-ocf": [
```

```
            "precision = ocf.precision"
          ],
          "x-ocf-alias": "oic.r.temperature",
          "x-to-ocf": [
            "ocf.precision = precision"
          ]
        }
      },
      "updatemintime": {
        "type": "integer",
        "x-ocf-conversion": {
          "x-from-ocf": [
            "updatemintime = ocf.minnotifyperiod"
          ],
          "x-ocf-alias": "oic.r.value.conditional",
          "x-to-ocf": [
            "ocf.minnotifyperiod = updatemintime"
          ]
        }
      }
    },
    "required": [
      "currentvalue",
      "precision",
      "updatemintime"
    ],
    "type": "object"
  }

 }
}
```


A.7.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
currentvalue	oic.r.temperature	temperature = currentValueunits = C	oneOf	Measured value
updatemintime	oic.r.value.conditional	ocf.minnotifyperiod = updatemintime	updatemintime = ocf.minnotifyperiod	
precision	oic.r.temperature	ocf.precision = precision	precision = ocf.precision	

A.7.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CurrentTemperatureResURI		get			

A.8 Cycle Control Mapping

A.8.1 Introduction

This API defines the mapping between an instance of an AllJoyn CycleControl interface and the OCF OperationalState Resource.
The AllJoyn interface also supports a Method, ExecuteOperationalCommand; this is handled in OCF using an instance of oic.r.actuator within an oic.r.action collection. Please see Section 8 of the Mapping Specification for specifics.

A.8.2 Example URI

/CycleControlResURI

A.8.3 Resource Type

The resource type (rt) is defined as: ['oic.r.operationalstate'].

A.8.4 Swagger2.0 Definition


```
{
  "swagger": "2.0",
  "info": {
    "title": "Cycle Control Mapping",
    "version": "OCFv1.0.0-20170317",
    "license": {
      "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n        1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2.  Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n     HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/CycleControlResURI" : {
      "get": {
        "description": "This API defines the mapping between an instance of an AllJoyn CycleControl
interface and the OCF OperationalState Resource.\nThe AllJoyn interface also supports a Method,
ExecuteOperationalCommand; this is handled in OCF using an instance of oic.r.actuator within an
oic.r.action collection.\nPlease see Section 8 of the Mapping Specification for specifics.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-sensor"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt":     ["oic.r.operationalstate"]
                }
                ,
              "schema": { "$ref": "#/definitions/RetrieveSchema" }
            }
        }
      }
    }
  },
  "parameters": {
    "interface-sensor" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.s", "oic.if.baseline"]
    }
  },
  "definitions": {
    "RetrieveSchema" :
            {
        "properties": {
          "cyclephase": {
            "description": "Current phase of the operational cycle",
            "type": "integer",
```

```
            "x-ocf-conversion": {
              "x-from-ocf": [
                "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
                "cyclephase = indexof statearray[currentmachinestate[0]]"
              ],
              "x-ocf-alias": "oic.r.operationalstate",
              "x-to-ocf": [
                "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
                "currentmachinestate = phasearray[cyclephase]"
              ]
            }
          },
          "getvendorphasedescription": {
            "description": "Get cycle phases description",
            "x-ocf-conversion": {
              "x-ocf-alias": "oic.r.action"
            },
            "x-ocf-type": "method"
          },
          "supportedcyclephases": {
            "description": "Array of cycle phases supported by the Appliance.",
            "items": {
              "type": "integer"
            },
            "type": "array",
            "x-ocf-conversion": {
              "x-from-ocf": [
                "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
                "for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof
phasearray[machinestates[x]]"
              ],
              "x-ocf-alias": "oic.r.operationalstate",
              "x-to-ocf": [
                "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
                "for x=0, x < sizeof(supportedcyclephases): machinestates[x] =
phasearray[supportedcyclephases[x]]"
              ]
            }
          }
        },
        "required": [
          "cyclephase",
          "supportedcyclephases"
        ],
        "type": "object"
      }

  }
}
```


A.8.5 Property Definition

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	Description
supportedcyclephases	oic.r.operationalstate	phasearray = [Unavailable,Preheating,Cooking,Cleaning]for x=0, x < sizeof(supportedcyclephases): machinestates[x] = phasearray[supportedcyclephases[x]]	phasearray = [Unavailable,Preheating,Cooking,Cleaning]for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof phasearray[machinestates[x]]	Array of cycle phases supported by the Appliance.

cyclephase	oic.r.operationalstate	phasearray = [Unavailable,Preheating,Cooking,Cleaning]currentmachinestate = phasearray[cyclephase]	phasearray = [Unavailable,Preheating,Cooking,Cleaning]cyclephase = indexof statearray[currentmachinestate[0]]	Current phase of the operational cycle
getvendorphasedescription	oic.r.action			Get cycle phases description

A.8.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/CycleControlResURI		get			

A.9 Fan Speed Level Mapping

A.9.1 Introduction

This API defines the mapping between an instance of an AllJoyn FanSpeedLevel interface and an OCF AirFlow Resource.
Note that the setting of the FanSpeedLevel to '0x00' (off) is handled via the 'OffControl' interface rather than writing directly to this interface.
In such a case an instance of Binary Switch shall be exposed on the OCF side; this can be modeled as AirFlowControl which is then a collection of Binary Switch and AirFlow.

A.9.2 Example URI

/FanSpeedLevelResURI

A.9.3 Resource Type

The resource type (rt) is defined as: ['oic.r.airflow'].

A.9.4 Swagger2.0 Definition


```
{
  "swagger": "2.0",
  "info": {
    "title": "Fan Speed Level Mapping",
    "version": "OCFV1.0.0-20170317",
    "license": {
      "name": "Copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n        1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2.  Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGE.\n"
    }
  },
```

```
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/FanSpeedLevelResURI" : {
      "get": {
        "description": "This API defines the mapping between an instance of an AllJoyn
FanSpeedLevel interface and an OCF AirFlow Resource.\nNote that the setting of the FanSpeedLevel to
'0x00' (off) is handled via the 'OffControl' interface rather than writing directly to this
interface.\nIn such a case an instance of Binary Switch shall be exposed on the OCF side; this can
be modeled as AirFlowControl which is then a collection of Binary Switch and AirFlow.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-actuator"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt":     ["oic.r.airflow"]
                }
                ,
              "schema": { "$ref": "#/definitions/RetrieveSchema" }
            }
        }
      },
      "post": {
        "description": "",
        "parameters": [
          {"$ref": "#/parameters/interface-actuator"},
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": { "$ref": "#/definitions/UpdateSchema" }
          }
        ],
        "responses": {
            "200": {
              "description" : "",
              "schema": { "$ref": "#/definitions/UpdateSchema" }
            }
        }
      }
    }
  },
  "parameters": {
    "interface-actuator" :{
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.a", "oic.if.baseline"]
    }
  },
  "definitions": {
    "RetrieveSchema" :
        {
        "properties": {
          "automode": {
            "description": "Auto mode status.",
            "type": "integer",
            "x-ocf-conversion": {
              "x-from-ocf": [
                "automode = ocf.automode",
                "otherwise: automode = NotSupported(0xFF)"
              ],
              "x-ocf-alias": "oic.r.airflow",
              "x-to-ocf": [
                "if automode != NotSupported(0xFF)",
                " ocf.automode = automode",
                "else no mapping"
```