

OCF Resource to AllJoyn Interface Mapping

VERSION 1.0.0_Korean | June 2017



OPEN CONNECTIVITY
FOUNDATION™

CONTACT admin@openconnectivity.org

Copyright Open Connectivity Foundation, Inc. © 2016-2017.

All Rights Reserved.

법적 고지 사항

본 문서에 기재된 내용 중 그 어느 것도 명시적 또는 암시적으로 기재 내용에 있어서 어떠한 형태의 사용 허가를 부여하거나 본 문서의 작성자 또는 개발자 중 어느 누구가 소유 또는 관할하는 어떠한 지식재산에 대해 어떠한 형태의 사용 허가도 부여하는 것을 의미하지 않습니다. 여기에 포함된 정보는 “있는 그대로” 제공되며, 적용 가능한 법에 의해 허용되는 최대 한도까지 본 스펙의 작성자 및 개발자는 특정한 목적을 위한 판매 적격성 또는 적합성의 암시적 보증을 포함하지만 이에 한정되지 않는 명시적 또는 암시적인 성문법 또는 불문법 상의 기타 모든 보증 및 조건에 대해 일절 책임을 지지 않습니다. OPEN CONNECTIVITY FOUNDATION, INC.는 비침해, 정확성, 또는 바이러스 비 감염에 대한 모든 보증에 대해서도 일절 책임을 지지 않습니다.

OCF 로고는 미국 및 다른 국가에서 Open Connectivity Foundation, Inc 의 상표입니다. *그 밖의 명칭 및 상표는 해당하는 소유자의 자산일 수 있습니다.

Copyright © 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.

For Translation to Local Language

이들 저작물의 복사 또는 기타 형태의 복제 및/또는 배포는 엄격하게 금지되어 있습니다.

· 본 OCF 스펙 번역 버전은 OCF 기반의 제품 개발을 장려하고 이에 도움이 되도록 규범적인 영문 원본 버전으로부터 작성되었습니다. 영문 스펙의 정확한 번역을 위한 모든 노력을 기울이기는 하였지만 본 번역 버전을 규범적으로 간주해서는 안됩니다. OCF 인증 프로그램은 명백하게 영문 스펙을 기준으로 개발되어야 하며, 어떠한 면제 또는 면책 요구도 영문 스펙의 문구를 기준으로 평가되어야 합니다.

· 최신 영문 버전 스펙의 공개로부터 번역 버전의 공개까지는 소정의 지연이 있을 수 있습니다.

· OCF 스펙의 최신 영문 버전 및 해당 번역 버전에 관해서는

<https://openconnectivity.org/developer/specifications> 를 참조하여 주십시오.

22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51

목차

1	적용 범위.....	Error! Bookmark not defined.
2	인용 표준.....	12
3	용어, 정의, 기호 및 약어	13
3.1	용어 및 정의.....	13
3.2	기호 및 약어.....	13
3.3	협약.....	13
4	문서 규약 및 구성.....	13
4.1	표기법	13
4.2	Data type.....	14
5	동작 원리	14
5.1	상호 연동 접근	14
5.2	매핑 Syntax	15
5.2.1	일반	15
5.2.2	값 할당	15
5.2.3	Property 명명.....	15
5.2.4	배열.....	15
5.2.5	Default 매핑.....	15
5.2.6	조건부 매핑	16
5.2.7	루프	Error! Bookmark not defined.
5.2.8	메소드(Method) 실행.....	16
6	Device Type 매핑	16
6.1	개요	16
6.2	AllJoyn Device Type 대 OCF Device Type.....	17
6.3	AllJoyn 에 대응하지 않는 OCF Device Type	18
7	Resource 대 Interface 등가.....	19
7.1	개요	19
7.2	AllJoyn Interface 에서 OCF Resource 로 매핑	19

52	8	상세 매핑 API.....	21
53	8.1	Air Quality 매핑.....	23
54	8.1.1	개요.....	23
55	8.1.2	URI 예.....	23
56	8.1.3	Resource Type.....	23
57	8.1.4	RAML 정의.....	23
58	8.1.5	Property 정의.....	23
59	8.1.6	CRUDN 동작.....	25
60	8.2	Air Quality Level 매핑.....	26
61	8.2.1	개요.....	26
62	8.2.2	URI 예.....	26
63	8.2.3	Resource Type.....	26
64	8.2.4	RAML 정의.....	26
65	8.2.5	Property 정의.....	28
66	8.2.6	CRUDN 동작.....	29
67	8.3	Current Humidity 매핑.....	29
68	8.3.1	개요.....	29
69	8.3.2	URI 예.....	29
70	8.3.3	Resource Type.....	29
71	8.3.4	RAML 정의.....	29
72	8.3.5	Property 정의.....	30
73	8.3.6	CRUDN 동작.....	30
74	8.4	Current Temperature 매핑.....	31
75	8.4.1	개요.....	31
76	8.4.2	URI 예.....	31
77	8.4.3	Resource Type.....	31
78	8.4.4	RAML 정의.....	31
79	8.4.5	Property 정의.....	33
80	8.4.6	CRUDN 동작.....	33
81	8.5	Target Humidity 매핑.....	33
82	8.5.1	개요.....	33
83	8.5.2	URI 예.....	33
84	8.5.3	Resource Type.....	33

85	8.5.4	RAML 정의	33
86	8.5.5	Property 정의	38
87	8.5.6	CRUDN 동작	39
88	8.6	Target Temperature 매핑	39
89	8.6.1	개요	39
90	8.6.2	URI 예	39
91	8.6.3	Resource Type	39
92	8.6.4	RAML 정의	39
93	8.6.5	Property 정의	44
94	8.6.6	CRUDN 동작	44
95	8.7	Audio Volume 매핑	45
96	8.7.1	개요	45
97	8.7.2	URI 예	45
98	8.7.3	Resource Type	45
99	8.7.4	RAML 정의	45
100	8.7.5	Property 정의	48
101	8.7.6	CRUDN 동작	48
102	8.8	Climate Control Mode 매핑	48
103	8.8.1	개요	48
104	8.8.2	URI 예	49
105	8.8.3	Resource Type	49
106	8.8.4	RAML 정의	49
107	8.8.5	Property 정의	53
108	8.8.6	CRUDN 동작	53
109	8.9	Closed Status 매핑	53
110	8.9.1	개요	53
111	8.9.2	URI 예	53
112	8.9.3	Resource Type	53
113	8.9.4	RAML 정의	53
114	8.9.5	Property 정의	54
115	8.9.6	CRUDN 동작	55
116	8.10	Cycle Control 매핑	55
117	8.10.1	개요	55

118	8.10.2	URI 예	55
119	8.10.3	Resource Type	55
120	8.10.4	RAML 정의	55
121	8.10.5	Property 정의	57
122	8.10.6	CRUDN 동작	57
123	8.11	Fan Speed Level 매핑	57
124	8.11.1	개요	57
125	8.11.2	URI 예	57
126	8.11.3	Resource Type	57
127	8.11.4	RAML 정의	57
128	8.11.5	Property 정의	61
129	8.11.6	CRUDN 동작	61
130	8.12	Heating Zone 매핑	62
131	8.12.1	개요	62
132	8.12.2	URI 예	62
133	8.12.3	Resource Type	62
134	8.12.4	RAML 정의	62
135	8.12.5	Property 정의	63
136	8.12.6	CRUDN 동작	64
137	8.13	HVAC Fan Mode 매핑	64
138	8.13.1	개요	64
139	8.13.2	URI 예	64
140	8.13.3	Resource Type	64
141	8.13.4	RAML 정의	64
142	8.13.5	Property 정의	68
143	8.13.6	CRUDN 동작	68
144	8.14	On Off 매핑	68
145	8.14.1	개요	68
146	8.14.2	URI 예	68
147	8.14.3	Resource Type	68
148	8.14.4	RAML 정의	68
149	8.14.5	Property 정의	72
150	8.14.6	CRUDN 동작	72

151	8.15	Oven Cycle Phase 매핑	72
152	8.15.1	개요.....	72
153	8.15.2	URI 예	73
154	8.15.3	Resource Type	73
155	8.15.4	RAML 정의	73
156	8.15.5	Property 정의.....	74
157	8.15.6	CRUDN 동작.....	75
158		Annex A Swagger2.0 (단순 정보).....	76
159	A.1	Audio Volume 매핑	76
160	A.1.1	개요.....	76
161	A.1.2	URI 예	76
162	A.1.3	Resource Type	76
163	A.1.4	Swagger2.0 정의.....	76
164	A.1.5	Property 정의.....	79
165	A.1.6	CRUDN 동작.....	79
166	A.2	Climate Control Mode 매핑.....	79
167	A.2.1	개요.....	79
168	A.2.2	URI 예	79
169	A.2.3	Resource Type	79
170	A.2.4	Swagger2.0 정의.....	79
171	A.2.5	Property 정의.....	82
172	A.2.6	CRUDN 동작.....	82
173	A.3	Closed Status 매핑	82
174	A.3.1	개요.....	82
175	A.3.2	URI 예	82
176	A.3.3	Resource Type	82
177	A.3.4	Swagger2.0 정의.....	82
178	A.3.5	Property 정의.....	84
179	A.3.6	CRUDN 동작.....	85
180	A.4	Air Quality 매핑	85
181	A.4.1	개요.....	85
182	A.4.2	URI 예	85
183	A.4.3	Resource Type	85

184	A.4.4	Swagger2.0 정의.....	85
185	A.4.5	Property 정의.....	87
186	A.4.6	CRUDN 동작.....	88
187	A.5	Air Quality Level 매핑	88
188	A.5.1	개요.....	88
189	A.5.2	URI 예	88
190	A.5.3	Resource Type	88
191	A.5.4	Swagger2.0 정의.....	89
192	A.5.5	Property 정의.....	90
193	A.5.6	CRUDN 동작.....	91
194	A.6	Current Humidity 매핑	92
195	A.6.1	개요.....	92
196	A.6.2	URI 예	92
197	A.6.3	Resource Type	92
198	A.6.4	Swagger2.0 정의.....	92
199	A.6.5	Property 정의.....	93
200	A.6.6	CRUDN 동작.....	94
201	A.7	Current Temperature 매핑	94
202	A.7.1	개요.....	94
203	A.7.2	URI 예	94
204	A.7.3	Resource Type	94
205	A.7.4	Swagger2.0 정의.....	94
206	A.7.5	Property 정의.....	96
207	A.7.6	CRUDN 동작.....	96
208	A.8	Cycle Control 매핑	97
209	A.8.1	개요.....	97
210	A.8.2	URI 예	97
211	A.8.3	Resource Type	97
212	A.8.4	Swagger2.0 정의.....	97
213	A.8.5	Property 정의.....	99
214	A.8.6	CRUDN 동작.....	99
215	A.9	Fan Speed Level 매핑	99
216	A.9.1	개요.....	99

217	A.9.2	URI 예	99
218	A.9.3	Resource Type	100
219	A.9.4	Swagger2.0 정의.....	100
220	A.9.5	Property 정의.....	102
221	A.9.6	CRUDN 동작.....	103
222	A.10	Heating Zone 매핑	103
223	A.10.1	개요.....	103
224	A.10.2	URI 예	103
225	A.10.3	Resource Type	103
226	A.10.4	Swagger2.0 정의.....	103
227	A.10.5	Property 정의.....	105
228	A.10.6	CRUDN 동작.....	106
229	A.11	HVAC Fan Mode 매핑	106
230	A.11.1	개요.....	106
231	A.11.2	URI 예	106
232	A.11.3	Resource Type	106
233	A.11.4	Swagger2.0 정의.....	106
234	A.11.5	Property 정의.....	109
235	A.11.6	CRUDN 동작.....	109
236	A.12	On Off 매핑.....	109
237	A.12.1	개요.....	109
238	A.12.2	URI 예	110
239	A.12.3	Resource Type	110
240	A.12.4	Swagger2.0 정의.....	110
241	A.12.5	Property 정의.....	112
242	A.12.6	CRUDN 동작.....	112
243	A.13	Oven Cycle Phase 매핑	112
244	A.13.1	개요.....	112
245	A.13.2	URI 예	112
246	A.13.3	Resource Type	112
247	A.13.4	Swagger2.0 정의.....	112
248	A.13.5	Property 정의.....	114
249	A.13.6	CRUDN 동작.....	115

250	A.14	Target Humidity 매핑	115
251	A.14.1	개요.....	115
252	A.14.2	URI 예	115
253	A.14.3	Resource Type	115
254	A.14.4	Swagger2.0 정의.....	115
255	A.14.5	Property 정의.....	119
256	A.14.6	CRUDN 동작.....	119
257	A.15	Target Temperature 매핑.....	120
258	A.15.1	개요.....	120
259	A.15.2	URI 예	120
260	A.15.3	Resource Type	120
261	A.15.4	Swagger2.0 정의.....	120
262	A.15.5	Property 정의.....	124
263	A.15.6	CRUDN 동작.....	124
264			
265			

266

도면

267

No table of figures entries found.

268	표	
269	표 6-1 AllJoyn 에서 OCF Device Type 으로의 매핑.	17
270	표 6-1 AllJoyn EquivalentOCF Device Name 없는 OCF Device Type	18
271	표 7-1 AllJoyn Interface 에서 OCF Resource Type 으로의 매핑 – 최소 Interface 집합.....	19
272	표 7-2 AllJoyn Interface 에서 OCF Resource Type 으로의 매핑 – 옵션 Interface 집합.....	20
273	표 8-1 Interface 와 Resource 요약	22
274		

1 적용 범위

OCF Resource로부터 AllJoyn Interface에의 매핑 스펙 (“본 스펙”)은 AllJoyn 정의 Interface와 OCF 정의 Resource 간의 상호 연동을 제공하기 위한 상세 매핑 정보를 제공한다.

본 스펙은 Device Type에 대한 (AllJoyn에서 OCF로 또는 OCF에서 AllJoyn으로) 매핑을 제공하고, 필수 및 옵션 AllJoyn interface에 대한 등가 OCF Resource를 식별하고, 각 interface에 대해 매핑을 표준적으로 정의하기 위해 JSON schema에 대한 OCF 정의 확장을 사용한 상세 Property 간 매핑을 정의한다.

2 인용 표준

다음의 문헌은, 일부 또는 전부가 본 스펙에서 표준적으로 인용되며 어플리케이션에 있어서 필수적이다. 날짜가 표기된 문헌의 경우에는 인용된 판만 적용된다. 날짜가 표기되지 않은 문헌의 경우에는 참조된 문헌의 최신판(보정 내용 포함)이 적용된다.

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/>

3 용어, 정의, 기호 및 약어

3.1 용어 및 정의

3.2 기호 및 약어

3.2.1

OCF

Open Connectivity Foundation

본 스펙을 제작한 기구.

3.2.2

RAML

RESTful API Modelling Language

RAML 은 사실상 RESTful API 를 기술하는 간단명료한 방식이다. RAML 참조.

3.3 협약

본 스펙에서, 다수의 용어, 조건, 메커니즘, 시퀀스, 파라미터, 이벤트, 상태, 또는 유사한 용어는 각 단어의 첫 번째 문자를 대문자로 표기하고 나머지는 소문자로 표기한다 (예: Network Architecture). 이러한 단어가 소문자로 표기되었을 때는 일반적인 기술적 영어의 의미를 갖는다.

4 문서 규약 및 구성

본 스펙을 위해 OCF Core 스펙 및 OCF Resource Type 스펙 내의 용어 및 정의가 적용된다.

4.1 표기법

본 스펙에서 기능은 다음과 같이 필수(Required), 권고(Recommended), 허가(Allowed), 또는 사용 금지(DEPRECATED)로 분류된다.

필수 (강제 또는 의무적)

이러한 기본 기능은 Mapping 스펙을 준수하도록 구현되어야 한다. "하지 말아야 한다"나 "금지한다" 등의 구절은 금지되는, 즉, 수행하는 경우 구현이 스펙을 준수하지 않음을 의미하는 행위를 나타낸다.

권고 (또는 제안)

이러한 기능은 Mapping 스펙에 의해 지원되는 기능을 부가하며 구현되어야 한다. 권고 기능은, 통상적으로 중대한 복잡성의 증가 없이 Mapping 스펙의 기능을 이용한다. 규정 준수 테스트를 위해 권고 기능이 구현된다면 이 가이드라인에 따르는 특정 요건을 만족해야 한다. 일부 권고 기능은 추후에 필수 요건이 될 수 있다. "하지 않는 것이 좋다"라는 표현은 허용되지만 권고하지 않는 행위를 나타낸다.

338
339
340 허가 (또는 허용)
341 이러한 기능은 Mapping 스펙에 의해 필수적이지도 않을 뿐더러 권고되지도 않지만 기능이
342 구현된다면 이 가이드라인에 따르는 특정 요건을 만족해야 한다.

343
344 조건부 허용 (CA)
345 정의 또는 행위는 조건에 의존한다. 특정 조건이 만족되면 정의 또는 행위가 허용되고 그렇지
346 않으면 허용되지 않는다.

347 조건부 필수 (CR)
348 정의 또는 행위는 조건에 의존한다. 특정 조건이 만족되면 정의 또는 행위가 필수로 된다.
349 그렇지 않으면 특별한 기재가 없는 한 default 로 허용된다.

350
351 사용 금지
352 이에 해당하는 기능은 본 스펙에서 설명은 하고 있지만 역 호환성을 제외하고는 구현되어서는
353 안된다. 현재 스펙에 따르는 동작 동안 사용 금지된 기능의 발생은 구현 동작에 어떤 영향도
354 끼치지 않으며 어떠한 에러 상태도 생성하지 않는다. 역 호환성은 기능이 구현되고 특정된 대로
355 기능할 것을 요구하지만 본 스펙에 따르는 구현에 의해 사용되어서는 안된다.

356
357
358 문자 그대로 해석되는 String 은 "인용부호"를 사용한다.
359 강조하는 단어는 *이탤릭체*로 표기한다.

360 **4.2 Data type**
361 OCF Core 스펙 참조.

362 5 동작 원리

363 5.1 상호 연동 접근

364 AllJoyn 정의 interface 와 OCF 정의 Resource Type 간의 상호 연동은 상호 운용성을 위한 Derived
365 Model 에 기술된 도출 모델 구문을 사용해서 모델링 된다. OCF data model 내에 등가성이 요구되는
366 AllJoyn interface 의 최소 집합의 결정은 AllJoyn 내의 CDM Project 에 의해 정의되는 각 device

367 type 에 대해 요구되는 interface 를 나열함으로써 수행된다. AllJoyn interface 가 방법을 지원하면
368 작동 디자인 패턴이 적용된다.

369

370 5.2 매핑 Syntax

371 본 스펙에 사용되는 도출된 모델링에 대해 정의된 구문론 내에는 실제 Property-Property 등가 또는
372 매핑을 정의하는 두 개의 블록이 있다. 이들 블록은 'x-to-ocf'와 'x-from-ocf'의 키워드에 의해
373 정의된다. 상호 운용성을 위한 도출된 모델은 이들 블록에 대한 엄격한 syntax 를 정의하지 않아서
374 pseudo-code 된 매핑 로직을 포함하는 자유 형식의 스트링 배열이 된다.

375 본 스펙 내에서는 정의된 매핑 로직의 일관성, 재 사용성, 및 확장성을 확보하기 위해 이들 블록에
376 다음의 서브 섹션의 규칙을 적용한다.

377

378 5.2.1 일반

379 모든 서술은 개행 복귀로 종료된다.

380 5.2.2 값 할당

381 하나의 값을 다른 것에 할당할 때 등호 (=)를 사용한다. 할당 받는 쪽이 연산자의 왼쪽에 위치하고
382 할당되는 값이 오른쪽에 위치한다.

383 5.2.3 Property 명명

384 모든 Property 명칭은 원래 모델에 의해 사용되는 명칭과 동일하다. 예를 들어, OCF Temperature
385 Resource 에 정의된 'temperature'라는 property 는 도출된 생태계(AllJoyn)에서도 구문적으로
386 동등한 Property 명칭이 사용된다.

387 구문적으로 동등한 값에 대해 OCF 와 도출된 생태계 양쪽에 의해 동일한 명칭이 사용되면 애매함을
388 피하기 위해 OCF 정의 Property 의 명칭 앞에 생태계 지정자 'ocf'가 붙는다 (예: 'ocf.step').

389

390 5.2.4 배열

391 배열 요소는, 예를 들어, range[1]과 같이 사각 괄호 '[' 안에 요소의 색인을 포함해서 나타낸다. 모든
392 배열은 인덱스 0 에서 시작한다. 전체 배열을 참조할 때는, 예를 들어, selectablehumiditylevels[]와
393 같이 인덱스를 포함하지 않는다.

394 5.2.5 Default 매핑

395 매핑되는 하나 이상의 Property 가 소스 모델에서 옵션이어서 지정된 매핑을 할 수 없는 경우가 있다.
396 그러한 경우에는 default 매핑이 수행된다. default 매핑은 할당 앞에 수식어 'otherwise:'를 붙여서
397 표현한다 (예: 'otherwise: step = 1').

398

399 5.2.6 조건부 매핑

400 매핑이 다른 조건의 만족 여부에 의존하는 경우 다음과 같은 구문이 적용된다.

401

402 if ‘condition’, ‘mapping’.

403 예: if step >0, ocf.step = step.

404 5.2.7 루프

405 매핑을 소정의 조건에 따라 반복되는 루프로 표현할 수 있는 경우 다음과 같은 구문이 적용된다.

406

407

408 ‘initialize’는 초기 로컬 루프 제어 변수 설정을 나타내고,

409 ‘condition’은 루프 컨트롤러로, 조건이 “거짓”이 될 때까지 루프가 반복됨을 나타내고,

410 ‘increment’는 제어 변수의 갱신을 나타내며, 생략 시에는 ‘1’씩 증가한다.

411

412

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

414

415 5.2.8 메소드(Method) 실행

416 Method 또는 OCF Resource 로부터의 매핑의 일환으로 도출된 생태계로부터의 remote procedure
417 call (RPC)의 실행은 사용 가능한 resource, service, interface, 또는 그 밖의 구성 식별자 및 방법
418 또는 RPC 명칭 간에 이중 콜론 ‘::’ 구분 문자를 사용한다. Method 명칭은 항상 뒤에 임의의
419 파라미터를 포함하는 괄호가 뒤따른다.

420 예를 들어, AllJoyn 으로부터의 switchon() Method 을 처리할 때 operation.oncontrol::switchon()과
421 같이 완전한 Method 적용을 제공한다.

422

423 6 Device Type 매핑

424 6.1 개요

425 이 섹션은 Device Type 으로/으로부터의 매핑을 다룬다.

6.2 AllJoyn Device Type 에서 OCF Device Type 으로의 매핑

다음의 표는 AllJoyn 정의 Device Type (AllJoyn Common Data Model Interface 정의 참조)와 OCF 정의 Device Types (OCF Smart Home Device 스펙의 표 10-1 참조) 간의 등가 매핑을 보여준다. AllJoyn 정의에 대한 최소 interface 집합은 HAE Theory of Operation 에 제공되고, 각 OCF Device 에 대한 최소 Resource 집합은 OCF Smart Home Device 스펙에 제공된다.

표 6-1 AllJoyn 에서 OCF Device Type 으로의 매핑

분류	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 (Resource 에 매핑된다)
	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

433

434 6.3 AllJoyn 에 대응하지 않는 OCF Device Type

435 다음의 표는 AllJoyn 내에 직접적으로 등가 항목이 없는 OCF 정의 Device Type 을 보여주며, 이들은
436 AllJoyn Device Type 의 형태가 'Other' (Id '1')에 매핑된다.

437

표 6-2 AllJoyn 등가가 없는 OCF 디바이스 타입

OCF 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 대 Interface 등가

7.1 개요

이 섹션에서는 사용 가능한 AllJoyn Interface 의 완전한 집합을 나열하고, Interface 가 매핑되는 등가 OCF Resource Type 을 제공한다.

7.2 AllJoyn Interface 에서 OCF Resource 로 매핑

다음의 표는 AllJoyn 정의 Interface (AllJoyn Common Data Model Interface 정의 참조)와 OCF 정의 Resource Types (OCF Resource Type 스펙 참조) 간의 등가 매핑을 보여준다. Property 매핑에 의한 상세 Property 는 섹션 8 에 제공된다.

표 7-1 AllJoyn Interface 대 OCF Resource Type 매핑 – 최소 Interface 집합은 AllJoyn Device 를 위한 최소 집합의 일부 Interface 에 대한 매핑을 보여준다.

표 7-2 AllJoyn Interface 대 OCF Resource Type 매핑 – 옵션 Interface 집합은 AllJoyn Device 을 위한 옵션 Interface 에 대한 매핑을 보여준다. 도출된 모델링을 통한 이러한 interface 의 세부 변환(Deep Translation)은 이번 릴리즈의 스펙의 적용 범위에서 벗어난다.

표 7-1 AllJoyn Interface 에서 OCF Resource Type 로 매핑 – 최소 Interface 집합

AllJoyn Interface	OCF Resource Type Name	OCF Resource Type ID	OCF Interface
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	oic.r.audio	oic.if.a

	Controls		
Operation.Channel	매핑되지 않음		
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

452

표 7-2 AllJoyn Interface 에서 OCF Resource Type 으로 매핑 – 옵션 Interface 집합

453

AllJoyn Interface	OCF Resource Type Name	OCF Resource Type ID	OCF Interface
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	oic.r.media.input	oic.if.a

	Source List		
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 상세 매핑 API

이 섹션은 적용 범위 내의 모든 Interface 와 Resource 에 대해 API (RAML 사용)와 매핑 설명 ([Derived Model White Paper]에 기술된 도출된 모델링 구문에 따르는 JSON 사용)을 제공한다.

Annex A 는 RAML 과 JSON 대신에 Swagger2.0 을 사용하는 모든 매핑에 대한 정의를 제공한다.

표 8-1 Interface to Resource 요약은 Interface 서브 섹션에 대한 참조 및 링크를 제공한다.

표 8-1 Interface to Resource 요약

AllJoyn Interface Name	Equivalent Resource	매핑 섹션
Environment.CurrentAirQuality	oic.r.airqualitycollection	8.1
Environment.CurrentAirQualityLevel	oic.r.airqualitycollection	8.2
Environment.CurrentHumidity	oic.r.humidity	8.3
Environment.CurrentTemperature	oic.r.temperature	8.4
Environment.TargetHumidity	oic.r.humidity, oic.r.selectablelevels	8.5
Environment.TargetTemperature	oic.r.temperature	8.6
Operation.AudioVolume	oic.r.audio	8.7
Operation.ClimateControlMode	oic.r.mode, oic.r.operationalstate	8.8
Operation.ClosedStatus	oic.r.door	8.9
Operation.CycleControl	oic.r.operational.state	8.10
Operation.FanSpeedLevel	oic.r.airflow	8.11
Operation.HeatingZone	oic.r.heatingzonecollection	8.12
Operation.HvacFanMode	oic.r.mode	8.13
Operation.OnOffStatus, Operation.OnControl, Operation.OffControl	oic.r.switch.binary	8.14
Operation.OvenCyclePhase	oic.r.operationalstate	8.15

466

467 8.1 Air Quality 매핑

468 8.1.1 개요

469 이 API 는 AllJoyn AirQuality interface 와 OCF AirQuality Resource 간의 매핑을 정의한다.
470 복수의 AllJoyn AirQuality interface 의 인스턴스가 있는 경우, 각각의 인스턴스가 OCF AirQuality
471 Resource 의 인스턴스에 매핑된다. schema 내에 정의된 매핑은 OCF AirQuality Resource 의 개체를
472 설명한다. OCF AirQuality Resource 의 단일 인스턴스만 있더라도, OCF AirQualityCollection 의
473 인스턴스 내에 포함되어야 한다. collection 내의 링크 수는 노출되는 AllJoyn CurrentAirQuality
474 interface 의 인스턴스의 수와 같다. OCF 로부터의 매핑 시에 Resource 의 valueType 은
475 introspection 되며, 이 API 는 'Measured'로 설정되었을 때만 적용된다.

476

477

478 8.1.2 URI 예

479 /CurrentAirQualityResURI

480 8.1.3 Resource Type

481 resource type (rt)는 oic.r.airqualitycollection 으로 정의된다.

482 8.1.4 RAML 정의

```
483 #%RAML 0.8
484 title: CurrentAirQualityInterfaceMapping
485 version: OCFv1.0.0-20170317
486 traits:
487   - interface-sensor :
488     queryParameters:
489       if:
490         enum: ["oic.if.s", "oic.if.baseline"]
491
492 /CurrentAirQualityResURI:
493   description: |
494     This API defines the mapping between the AllJoyn AirQuality interface and the OCF AirQuality
495     Resource.
496     If more than one instance of the AirQuality interface is exposed then each instance maps to an
497     instance of the OCF AirQuality Resource.
498     The mapping defined in the schema describes the population of the OCF AirQuality Resource.
499     Even if there is only a single instance of an OCF AirQuality Resource this shall be included in
500     an instance of an OCF AirQualityCollection.
501     The number of links in the collection equates to the number of instances of the AllJoyn
502     CurrentAirQuality interface that are exposed.
503     When mapping from OCF the valueType of the Resource shall be introspected, this API is invoked
504     only if this is set to 'Measured'
505
506   is : ['interface-sensor']
507   get:
508     responses :
509       200:
510         body:
511           application/json:
512             schema: /
```



```

513     {
514         "id":
515         "http://openinterconnect.org/asamapping/schemas/asa.environment.currentairquality.json#",
516         "$schema": "http://json-schema.org/draft-04/schema#",
517         "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
518 reserved.",
519         "title": "Current Air Quality",
520         "definitions": {
521             "asa.environment.currentairquality": {
522                 "type": "object",
523                 "properties": {
524                     "contaminanttype": {
525                         "type": "integer",
526                         "description": "The contaminant type",
527                         "x-ocf-conversion": {
528                             "x-ocf-alias": "oic.r.airquality",
529                             "x-to-ocf": [
530                                 "valuetype = Measured",
531                                 "contaminanttypearray = [CH2O,CO2,CO,PM2_5,PM10,VOC]",
532                                 "ocf.contaminanttype = contaminanttypearray[contaminanttype]"
533                             ],
534                             "x-from-ocf": [
535                                 "contaminanttype = indexof contaminanttypearray[ocf.contaminanttype]"
536                             ]
537                         }
538                     },
539                     "currentvalue": {
540                         "type": "number",
541                         "x-ocf-conversion": {
542                             "x-ocf-alias": "oic.r.airquality",
543                             "x-to-ocf": [
544                                 "contaminantvalue = currentvalue"
545                             ],
546                             "x-from-ocf": [
547                                 "currentvalue = contaminantvalue"
548                             ]
549                         }
550                     },
551                     "minvalue": {
552                         "type": "number",
553                         "x-ocf-conversion": {
554                             "x-ocf-alias": "oic.r.airquality",
555                             "x-to-ocf": [
556                                 "range[0] = minvalue"
557                             ],
558                             "x-from-ocf": [
559                                 "minvalue = range[0]"
560                             ]
561                         }
562                     },
563                     "maxvalue": {
564                         "type": "number",
565                         "x-ocf-conversion": {
566                             "x-ocf-alias": "oic.r.airquality",
567                             "x-to-ocf": [
568                                 "range[1] = maxvalue"
569                             ],
570                             "x-from-ocf": [
571                                 "maxvalue = range[1]"
572                             ]
573                         }
574                     },
575                     "precision": {
576                         "type": "number",
577                         "x-ocf-conversion": {
578                             "x-ocf-alias": "oic.r.airquality",
579                             "x-to-ocf": [
580                                 "ocf.precision = precision"
581                             ],
582                             "x-from-ocf": [
583                                 "precision = ocf.precision"

```

```

584     ]
585     },
586     },
587     "updatemintime": {
588         "type": "integer",
589         "x-ocf-conversion": {
590             "x-ocf-alias": "oic.r.value.conditional",
591             "x-to-ocf": [
592                 "ocf.minnotifyperiod = updatemintime"
593             ],
594             "x-from-ocf": [
595                 "updatemintime = ocf.minnotifyperiod"
596             ]
597         }
598     }
599 }
600 },
601 },
602 "type": "object",
603 "allOf": [
604     { "$ref": "#/definitions/asa.environment.currentairquality" }
605 ],
606 "required":
607 [ "contaminanttype", "currentvalue", "minvalue", "maxvalue", "precision", "updatemintime" ]
608 }
609
610 example: /
611 {
612     "rt":      [ "oic.r.airqualitycollection" ]
613 }
614

```

8.1.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
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]	오염물질 유형

8.1.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/CurrentAirQualityResURI		get			

8.2 AirQuality Level 매핑

8.2.1 개요

이 API 는 AllJoyn AirQualityLevel interface 와 OCF AirQuality Resource 간의 매핑을 정의한다. 복수의 AirQualityLevel interface 인스턴스가 있는 경우, 이들 인스턴스는 각각 OCF AirQuality Resource 의 인스턴스에 매핑된다. schema 내에 정의된 매핑은 OCF AirQuality Resource 의 개체를 설명한다. OCF AirQuality Resource 의 단일 인스턴스만 있더라도 OCF AirQualityCollection 의 인스턴스 내에 포함되어야 한다. collection 내의 링크 수는 노출되는 AllJoyn CurrentAirQuality interface 의 인스턴스의 수와 같다. OCF 로부터의 매핑 시에 Resource 의 valueType 는 introspection 되며, 이 API 는 'Qualitative'로 설정되었을 때만 적용된다.

8.2.2 URI 예

/CurrentAirQualityLevelResURI

8.2.3 Resource Type

resource type (rt)는 oic.r.airqualitycollection 으로 정의된다.

8.2.4 RAML 정의

```
##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":
```

```

665 "http://openinterconnect.org/asamapping/schemas/asa.environment.currentairqualitylevel.json#",
666 "$schema": "http://json-schema.org/draft-04/schema#",
667 "description": "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
668 reserved.",
669 "title": "Current Air Quality Level",
670 "definitions": {
671   "asa.environment.currentairqualitylevel": {
672     "type": "object",
673     "properties": {
674       "contaminanttype": {
675         "type": "integer",
676         "description": "The contaminant type",
677         "x-ocf-conversion": {
678           "x-ocf-alias": "oic.r.airquality",
679           "x-to-ocf": [
680             "valuetype = Qualitative",
681             "if contaminanttype = 0, ocf.contaminanttype = CH2O",
682             "if contaminanttype = 1, ocf.contaminanttype = CO2",
683             "if contaminanttype = 2, ocf.contaminanttype = CO",
684             "if contaminanttype = 3, ocf.contaminanttype = PM2_5",
685             "if contaminanttype = 4, ocf.contaminanttype = PM10",
686             "if contaminanttype = 5, ocf.contaminanttype = VOC",
687             "if contaminanttype = 253, ocf.contaminanttype = Smoke",
688             "if contaminanttype = 254, ocf.contaminanttype = Odor",
689             "if contaminanttype = 255, ocf.contaminanttype = AirPollution"
690           ],
691           "x-from-ocf": [
692             "if ocf.contaminanttype = CH2O, contaminanttype = 0",
693             "if ocf.contaminanttype = CO2, contaminanttype = 1",
694             "if ocf.contaminanttype = CO, contaminanttype = 2",
695             "if ocf.contaminanttype = PM2_5, contaminanttype = 3",
696             "if ocf.contaminanttype = PM10, contaminanttype = 4",
697             "if ocf.contaminanttype = VOC, contaminanttype = 5",
698             "if ocf.contaminanttype = Smoke, contaminanttype = 253",
699             "if ocf.contaminanttype = Odor, contaminanttype = 254",
700             "if ocf.contaminanttype = AirPollution, contaminanttype = 255"
701           ]
702         }
703       },
704       "currentlevel": {
705         "type": "integer",
706         "x-ocf-conversion": {
707           "x-ocf-alias": "oic.r.airquality",
708           "x-to-ocf": [
709             "contaminantvalue = currentlevel"
710           ],
711           "x-from-ocf": [
712             "currentlevel = contaminantvalue"
713           ]
714         }
715       },
716       "maxlevel": {
717         "type": "integer",
718         "x-ocf-conversion": {
719           "x-ocf-alias": "oic.r.airquality",
720           "x-to-ocf": [
721             "range[0] = 0",
722             "range[1] = maxvalue"
723           ],
724           "x-from-ocf": [
725             "maxvalue = range[1]"
726           ]
727         }
728       }
729     }
730   },
731   "type": "object",
732   "allOf": [
733     {"$ref": "#/definitions/asa.environment.currentairqualitylevel"}
734   ],
735 }

```

```

736         "required": ["contaminanttype", "currentlevel", "maxlevel"]
737     }
738
739     example: /
740     {
741         "rt":      ["oic.r.airqualitycollection"]
742     }
743

```

744 8.2.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
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	오염물질 유형

745 8.2.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/CurrentAirQualityLevelResURI		get			

746 8.3 Humidity 매핑

747 8.3.1 개요

748 이 API 는 Sensor interface 에 있는 OCF Humidity 의 인스턴스와 AllJoyn Current Humidity
749 interface 간의 매핑을 정의한다. 습도 센서 상의 RETRIEVE 가 Environment.CurrentHumidity
750 Interface 의 인스턴스 상의 동작에 매핑된다.

751 8.3.2 URI 예

752 /CurrentHumidityResURI

753 8.3.3 Resource Type

754 resource type (rt)는 oic.r.humidity 로 정의된다.

755 8.3.4 RAML 정의

```

756 #%RAML 0.8
757 title: CurrentHumidityInterfaceMapping
758 version: OCFv1.0.0-20170317
759 traits:
760   - interface-sensor :
761       queryParameters:
762         if:
763           enum: ["oic.if.s", "oic.if.baseline"]
764
765 /CurrentHumidityResURI:
766   description: |
767     This API defines the mapping between an instance of an OCF Humidity which exposes only a sensor
768     interface
769     and the AllJoyn Current Humidity interface.
770     A RETRIEVE on a Temperature Sensor maps to an action on an instance of an
771     Environment.CurrentTemperature Interface.
772
773   is : ['interface-sensor']
774   get:
775     responses :
776       200:
777         body:
778           application/json:
779             schema: /
780             {
781               "id":
782               "http://openinterconnect.org/asamapping/schemas/asa.environment.currenthumidity.json#",
783               "$schema": "http://json-schema.org/draft-04/schema#",
784               "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
785               reserved.",
786               "title": "Current Humidity",
787               "definitions": {
788                 "asa.environment.currenthumidity": {
789                   "type": "object",
790                   "properties": {
791                     "currentvalue": {

```

```

792         "type": "number",
793         "description": "Measured value",
794         "x-ocf-conversion": {
795             "x-ocf-alias": "oic.r.humidity",
796             "x-to-ocf": [
797                 "humidity = currentValue"
798             ],
799             "x-from-ocf": [
800                 "currentvalue = humidity"
801             ]
802         },
803     },
804     "maxvalue": {
805         "type": "number",
806         "description": "Max measured value for humidity",
807         "x-ocf-conversion": {
808             "x-ocf-alias": "oic.r.humidity",
809             "x-to-ocf": [
810                 "range[0] = 0",
811                 "range[1] = maxvalue"
812             ],
813             "x-from-ocf": [
814                 "maxvalue = range[1]"
815             ]
816         },
817     },
818 },
819 },
820 },
821 "type": "object",
822 "allof": [
823     { "$ref": "#/definitions/asa.environment.currenthumidity" }
824 ],
825 "required": [ "currentvalue", "maxvalue" ]
826 }
827
828 example: /
829 {
830     "rt": [ "oic.r.humidity" ]
831 }
832

```

8.3.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
currentvalue	oic.r.humidity	humidity currentValue	currentvalue humidity	측정치
maxvalue	oic.r.humidity	range[0] range[1] maxvalue	maxvalue range[1]	습도에 대한 최대 측정치

8.3.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/CurrentHumidityResURI		get			

8.4 Current Temperature 매핑

8.4.1 개요

이 API 는 센서 interface 에 있는 OCF Temperature 의 인스턴스와 AllJoyn Current Temperature interface 간의 매핑을 정의한다. 온도 센서 상의 RETRIEVE 가 Environment.CurrentTemperature Interface 의 인스턴스 상의 동작에 매핑된다.

840

841 8.4.2 URI 예

842 /CurrentTemperatureResURI

843 8.4.3 Resource Type

844 resource type (rt)는: oic.r.temperature 로 정의된다.

845 8.4.4 RAML 정의

```
846 ##RAML 0.8
847 title: CurrentTemperatureInterfaceMapping
848 version: OCFv1.0.0-20170317
849 traits:
850 - interface-sensor :
851     queryParameters:
852         if:
853             enum: ["oic.if.s", "oic.if.baseline"]
854
855 /CurrentTemperatureResURI:
856     description: |
857         This API defines the mapping between an instance of an OCF Temperature which exposes only a
858 sensor interface
859 and the AllJoyn Current Temperature interface.
860 A RETRIEVE on a Temperature Sensor maps to an action on an instance of an
861 Environment.CurrentTemperature Interface.
862
863 is : ['interface-sensor']
864 get:
865     responses :
866         200:
867             body:
868                 application/json:
869                     schema: /
870                         {
871                             "id":
872 "http://openinterconnect.org/asamapping/schemas/asa.environment.currenttemperature.json#",
873                             "$schema": "http://json-schema.org/draft-04/schema#",
874                             "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
875 reserved.",
876                             "title": "Current Temperature",
877                             "definitions": {
878                                 "asa.environment.currenttemperature": {
879                                     "type": "object",
880                                     "properties": {
881                                         "currentvalue": {
882                                             "type": "number",
883                                             "description": "Measured value",
884                                             "x-ocf-conversion": {
885                                                 "x-ocf-alias": "oic.r.temperature",
886                                                 "x-to-ocf": [
887                                                     "temperature = currentValue",
888                                                     "units = C"
889                                                 ],
890                                                 "x-from-ocf": {
891                                                     "oneOf": [
892                                                         {
893                                                             "properties": {
894                                                                 "units": "string",
895                                                                 "enum": ["C"]
896                                                             },
```



```

897         "x-from-ocf": [
898             "currentvalue = temperature"
899         ],
900     },
901     {
902         "properties": {
903             "units": "string",
904             "enum": ["F"]
905         },
906         "x-from-ocf": [
907             "currentvalue = (temperature-32)*5/9"
908         ],
909     },
910     {
911         "properties": {
912             "units": "string",
913             "enum": ["K"]
914         },
915         "x-from-ocf": [
916             "currentvalue = temperature-273.15"
917         ],
918     }
919 ]
920 }
921 }
922 },
923 "precision": {
924     "type": "number",
925     "x-ocf-conversion": {
926         "x-ocf-alias": "oic.r.temperature",
927         "x-to-ocf": [
928             "ocf.precision = precision"
929         ],
930         "x-from-ocf": [
931             "precision = ocf.precision"
932         ]
933     },
934 },
935 "updatemintime": {
936     "type": "integer",
937     "x-ocf-conversion": {
938         "x-ocf-alias": "oic.r.value.conditional",
939         "x-to-ocf": [
940             "ocf.minnotifyperiod = updatemintime"
941         ],
942         "x-from-ocf": [
943             "updatemintime = ocf.minnotifyperiod"
944         ]
945     },
946 },
947 }
948 },
949 },
950 "type": "object",
951 "allOf": [
952     { "$ref": "#/definitions/asa.environment.currenttemperature" }
953 ],
954 "required": [ "currentvalue", "precision", "updatemintime" ]
955 }
956
957 example: /
958 {
959     "rt": [ "oic.r.temperature" ]
960 }
961

```

8.4.5 Property 정의

['AllJoyn']	OCF Resource	To OCF	From OCF	설명
-------------	--------------	--------	----------	----

Property name				
currentvalue	oic.r.temperature	temperature = currentValueunits = C	oneOf	측정치
updatemintime	oic.r.value.conditional	ocf.minnotifyperiod = updatemintime	updatemintime = ocf.minnotifyperiod	
precision	oic.r.temperature	ocf.precision = precision	precision = ocf.precision	

8.4.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/CurrentTemperatureResURI		get			

8.5 Target Humidity 매핑

8.5.1 개요

이 API 는 AllJoyn TargetHumidity Interface 의 인스턴스와 OCF Resource 간의 매핑을 정의한다. Humidity Sensor 의 POST 가 Environment.TargetHumidity Interface 의 인스턴스의 동작에 매핑된다.

8.5.2 URI 예

/TargetHumidityResURI

8.5.3 Resource Type

resource type (rt)는 oic.r.humidity 로 정의된다.

8.5.4 RAML 정의

```

#%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: /

```

```

997         {
998             "id":
999             "http://openinterconnect.org/asamapping/schemas/asa.environment.targethumidity.json#",
1000             "$schema": "http://json-schema.org/draft-04/schema#",
1001             "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1002 reserved.",
1003             "title": "Target Humidity",
1004             "definitions": {
1005                 "asa.environment.targethumidity": {
1006                     "type": "object",
1007                     "properties": {
1008                         "targetvalue": {
1009                             "type": "number",
1010                             "description": "Measured value",
1011                             "x-ocf-conversion": {
1012                                 "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
1013                                 "x-to-ocf": [
1014                                     "if minvalue != maxvalue, ocf.desiredhumidity =
1015 targetvalue;ocf.targetlevel = selectablehumiditylevels[0].",
1016                                     "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
1017                                 ],
1018                                 "x-from-ocf": [
1019                                     "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
1020                                     "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
1021                                 ]
1022                             }
1023                         },
1024                         "minvalue": {
1025                             "type": "number",
1026                             "x-ocf-conversion": {
1027                                 "x-ocf-alias": "oic.r.humidity",
1028                                 "x-to-ocf": [
1029                                     "range[0] = minvalue"
1030                                 ],
1031                                 "x-from-ocf": [
1032                                     "minvalue = range[0]",
1033                                     "otherwise: minvalue = 0"
1034                                 ]
1035                             }
1036                         },
1037                         "maxvalue": {
1038                             "type": "number",
1039                             "x-ocf-conversion": {
1040                                 "x-ocf-alias": "oic.r.humidity",
1041                                 "x-to-ocf": [
1042                                     "range[1] = maxvalue"
1043                                 ],
1044                                 "x-from-ocf": [
1045                                     "maxvalue = range[1]",
1046                                     "otherwise: maxvalue = 100"
1047                                 ]
1048                             }
1049                         },
1050                         "stepvalue": {
1051                             "type": "number",
1052                             "x-ocf-conversion": {
1053                                 "x-ocf-alias": "oic.r.humidity",
1054                                 "x-to-ocf": [
1055                                     "step = stepvalue"
1056                                 ],
1057                                 "x-from-ocf": [
1058                                     "stepvalue = step",
1059                                     "otherwise: stepvalue = 1"
1060                                 ]
1061                             }
1062                         },
1063                         "selectablehumiditylevels": {
1064                             "type": "array",
1065                             "items": {
1066                                 "type": "number"
1067                             }

```

```

1068         "x-ocf-conversion": {
1069             "x-ocf-alias": "oic.r.selectablelevels",
1070             "x-to-ocf": [
1071                 "availablelevels[] = selectablehumiditylevels[]"
1072             ],
1073             "x-from-ocf": [
1074                 "selectablehumiditylevels[] = availablelevels[]"
1075             ]
1076         }
1077     }
1078 }
1079 },
1080 },
1081 "type": "object",
1082 "allOf": [
1083     { "$ref": "#/definitions/asa.environment.targethumidity" }
1084 ],
1085 "required":
1086 [ "targetvalue", "minvalue", "maxvalue", "stepvalue", "selectablehumiditylevels" ]
1087 }
1088
1089     example: /
1090     {
1091         "rt":      ["oic.r.humidity", "oic.r.selectablelevels"]
1092     }
1093
1094     post:
1095     body:
1096     application/json:
1097     schema: /
1098     {
1099         "id":
1100 "http://openinterconnect.org/asamapping/schemas/asa.environment.targethumidity.json#",
1101         "$schema": "http://json-schema.org/draft-04/schema#",
1102         "description": "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1103 reserved.",
1104         "title": "Target Humidity",
1105         "definitions": {
1106             "asa.environment.targethumidity": {
1107                 "type": "object",
1108                 "properties": {
1109                     "targetvalue": {
1110                         "type": "number",
1111                         "description": "Measured value",
1112                         "x-ocf-conversion": {
1113                             "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
1114                             "x-to-ocf": [
1115                                 "if minvalue != maxvalue, ocf.desiredhumidity = targetvalue;ocf.targetlevel
1116 = selectablehumiditylevels[0].",
1117                                 "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
1118                             ],
1119                             "x-from-ocf": [
1120                                 "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
1121                                 "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
1122                             ]
1123                         }
1124                     },
1125                     "minvalue": {
1126                         "type": "number",
1127                         "x-ocf-conversion": {
1128                             "x-ocf-alias": "oic.r.humidity",
1129                             "x-to-ocf": [
1130                                 "range[0] = minvalue"
1131                             ],
1132                             "x-from-ocf": [
1133                                 "minvalue = range[0]",
1134                                 "otherwise: minvalue = 0"
1135                             ]

```

```

1136     }
1137   },
1138   "maxvalue": {
1139     "type": "number",
1140     "x-ocf-conversion": {
1141       "x-ocf-alias": "oic.r.humidity",
1142       "x-to-ocf": [
1143         "range[1] = maxvalue"
1144       ],
1145       "x-from-ocf": [
1146         "maxvalue = range[1]",
1147         "otherwise: maxvalue = 100"
1148       ]
1149     }
1150   },
1151   "stepvalue": {
1152     "type": "number",
1153     "x-ocf-conversion": {
1154       "x-ocf-alias": "oic.r.humidity",
1155       "x-to-ocf": [
1156         "step = stepvalue"
1157       ],
1158       "x-from-ocf": [
1159         "stepvalue = step",
1160         "otherwise: stepvalue = 1"
1161       ]
1162     }
1163   },
1164   "selectablehumiditylevels": {
1165     "type": "array",
1166     "items": {
1167       "type": "number"
1168     },
1169     "x-ocf-conversion": {
1170       "x-ocf-alias": "oic.r.selectablelevels",
1171       "x-to-ocf": [
1172         "availablelevels[] = selectablehumiditylevels[]"
1173       ],
1174       "x-from-ocf": [
1175         "selectablehumiditylevels[] = availablelevels[]"
1176       ]
1177     }
1178   }
1179 }
1180 }
1181 },
1182 "type": "object",
1183 "allOf": [
1184   {"$ref": "#/definitions/asa.environment.targethumidity"}
1185 ],
1186 "required":
1187 [ "targetvalue", "minvalue", "maxvalue", "stepvalue", "selectablehumiditylevels" ]
1188 }
1189
1190 responses :
1191 200:
1192   body:
1193     application/json:
1194       schema: /
1195         {
1196           "id":
1197 "http://openinterconnect.org/asamapping/schemas/asa.environment.targethumidity.json#",
1198           "$schema": "http://json-schema.org/draft-04/schema#",
1199           "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1200 reserved.",
1201           "title": "Target Humidity",
1202           "definitions": {
1203             "asa.environment.targethumidity": {

```

```

1204         "type": "object",
1205         "properties": {
1206             "targetvalue": {
1207                 "type": "number",
1208                 "description": "Measured value",
1209                 "x-ocf-conversion": {
1210                     "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
1211                     "x-to-ocf": [
1212                         "if minvalue != maxvalue, ocf.desiredhumidity =
1213 targetvalue;ocf.targetlevel = selectablehumiditylevels[0].",
1214                         "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
1215                     ],
1216                     "x-from-ocf": [
1217                         "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
1218                         "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
1219                     ]
1220                 }
1221             },
1222             "minvalue": {
1223                 "type": "number",
1224                 "x-ocf-conversion": {
1225                     "x-ocf-alias": "oic.r.humidity",
1226                     "x-to-ocf": [
1227                         "range[0] = minvalue"
1228                     ],
1229                     "x-from-ocf": [
1230                         "minvalue = range[0]",
1231                         "otherwise: minvalue = 0"
1232                     ]
1233                 }
1234             },
1235             "maxvalue": {
1236                 "type": "number",
1237                 "x-ocf-conversion": {
1238                     "x-ocf-alias": "oic.r.humidity",
1239                     "x-to-ocf": [
1240                         "range[1] = maxvalue"
1241                     ],
1242                     "x-from-ocf": [
1243                         "maxvalue = range[1]",
1244                         "otherwise: maxvalue = 100"
1245                     ]
1246                 }
1247             },
1248             "stepvalue": {
1249                 "type": "number",
1250                 "x-ocf-conversion": {
1251                     "x-ocf-alias": "oic.r.humidity",
1252                     "x-to-ocf": [
1253                         "step = stepvalue"
1254                     ],
1255                     "x-from-ocf": [
1256                         "stepvalue = step",
1257                         "otherwise: stepvalue = 1"
1258                     ]
1259                 }
1260             },
1261             "selectablehumiditylevels": {
1262                 "type": "array",
1263                 "items": {
1264                     "type": "number"
1265                 },
1266                 "x-ocf-conversion": {
1267                     "x-ocf-alias": "oic.r.selectablelevels",
1268                     "x-to-ocf": [
1269                         "availablelevels[] = selectablehumiditylevels[]"
1270                     ],
1271                     "x-from-ocf": [
1272                         "selectablehumiditylevels[] = availablelevels[]"
1273                     ]
1274                 }
1275             }
1276         }
1277     }

```

```

1275         }
1276     }
1277 }
1278 },
1279 "type": "object",
1280 "allOf": [
1281     { "$ref": "#/definitions/asa.environment.targethumidity" }
1282 ],
1283 "required":
1284 [ "targetvalue", "minvalue", "maxvalue", "stepvalue", "selectablehumiditylevels" ]
1285 }
1286
1287 example: /
1288 {
1289     "rt": [ "oic.r.humidity", "oic.r.selectablelevels" ]
1290 }
1291

```

1292 8.5.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
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.	측정 치
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	

1293 8.5.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/TargetHumidityResURI		get	post		

1294 8.6 Target Temperature 매핑

1295 8.6.1 개요

1296 이 API 는 sensor interface 에 있는 OCF Temperature 의 인스턴스와 AllJoyn Current Temperature
1297 interface 간의 매핑을 정의한다. 온도 센서 의 RETRIEVE 가 Environment.CurrentTemperature
1298 Interface 의 인스턴스의 동작에 매핑된다.

1299

1300 8.6.2 URI 예

1301 /TargetTemperatureResURI

1302 8.6.3 Resource Type

1303 resource type (rt)는 oic.r.temperature 로 정의된다.

1304 8.6.4 RAML 정의

```
1305 ##RAML 0.8
1306 title: TargetTemperatureInterfaceMapping
1307 version: OCFv1.0.0-20170317
1308 traits:
1309   - interface-actuator :
1310     queryParameters:
1311       if:
1312         enum: ["oic.if.a", "oic.if.baseline"]
1313
1314 /TargetTemperatureResURI:
1315   description: |
1316     This API defines the mapping between an instance of an OCF Temperature which exposes only a
1317     sensor interface
1318     and the AllJoyn Current Temperature interface.
1319     A RETRIEVE on a Temperature Sensor maps to an action on an instance of an
1320     Environment.CurrentTemperature Interface.
1321
1322   is : ['interface-actuator']
1323   get:
1324     responses :
1325       200:
1326         body:
1327           application/json:
1328             schema: /
1329               {
1330                 "id":
1331 "http://openinterconnect.org/asamapping/schemas/asa.environment.targettemperature.json#",
1332                 "$schema": "http://json-schema.org/draft-04/schema#",
1333                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1334 reserved.",
1335                 "title": "Target Temperature",
1336                 "definitions": {
1337                   "asa.environment.targettemperature": {
1338                     "type": "object",
1339                     "properties": {
1340                       "targetvalue": {
1341                         "type": "number",
1342                         "description": "Measured value",
1343                         "x-ocf-conversion": {
1344                           "x-ocf-alias": "oic.r.temperature",
1345                           "x-to-ocf": [
1346                             "temperature = targetvalue",
1347                             "units = C"
1348                           ],
1349                           "x-from-ocf": {
1350                             "oneOf": [
1351                               {
1352                                 "properties": {
1353                                   "units": "string",
1354                                   "enum": ["C"]
1355                                 },
1356                                 "x-from-ocf": [
1357                                   "targetvalue = temperature"
```



```

1358     ]
1359     },
1360     {
1361         "properties": {
1362             "units": "string",
1363             "enum": ["F"]
1364         },
1365         "x-from-ocf": [
1366             "targetvalue = (temperature-32)*5/9"
1367         ]
1368     },
1369     {
1370         "properties": {
1371             "units": "string",
1372             "enum": ["K"]
1373         },
1374         "x-from-ocf": [
1375             "targetvalue = temperature-273.15"
1376         ]
1377     }
1378 ]
1379 }
1380 }
1381 },
1382 "minvalue": {
1383     "type": "number",
1384     "x-ocf-conversion": {
1385         "x-ocf-alias": "oic.r.temperature",
1386         "x-to-ocf": [
1387             "range[0] = minvalue"
1388         ],
1389         "x-from-ocf": [
1390             "minvalue = range[0]",
1391             "otherwise: minvalue = -MAXINT"
1392         ]
1393     }
1394 },
1395 "maxvalue": {
1396     "type": "number",
1397     "x-ocf-conversion": {
1398         "x-ocf-alias": "oic.r.temperature",
1399         "x-to-ocf": [
1400             "range[1] = maxvalue"
1401         ],
1402         "x-from-ocf": [
1403             "maxvalue = range[1]",
1404             "otherwise: maxvalue = MAXINT"
1405         ]
1406     }
1407 },
1408 "step": {
1409     "type": "number",
1410     "x-ocf-conversion": {
1411         "x-ocf-alias": "oic.r.temperature",
1412         "x-to-ocf": [
1413             "ocf.step = step"
1414         ],
1415         "x-from-ocf": [
1416             "step = ocf.step",
1417             "otherwise: step = undefined (0x00)"
1418         ]
1419     }
1420 }
1421 }
1422 }
1423 },
1424 "type": "object",
1425 "allOf": [
1426     {"$ref": "#/definitions/asa.environment.targettemperature"}
1427 ],
1428 "required": [ "targetvalue", "minvalue", "maxvalue", "step" ]

```

```

1429         }
1430
1431     example: /
1432         {
1433             "rt":      ["oic.r.temperature"]
1434         }
1435
1436     post:
1437         body:
1438             application/json:
1439                 schema: /
1440                     {
1441                         "id":
1442 "http://openinterconnect.org/asamapping/schemas/asa.environment.targettemperature.json#",
1443 "$schema": "http://json-schema.org/draft-04/schema#",
1444 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1445 reserved.",
1446 "title": "Target Temperature",
1447 "definitions": {
1448     "asa.environment.targettemperature": {
1449         "type": "object",
1450         "properties": {
1451             "targetvalue": {
1452                 "type": "number",
1453                 "description": "Measured value",
1454                 "x-ocf-conversion": {
1455                     "x-ocf-alias": "oic.r.temperature",
1456                     "x-to-ocf": [
1457                         "temperature = targetvalue",
1458                         "units = C"
1459                     ],
1460                     "x-from-ocf": {
1461                         "oneOf": [
1462                             {
1463                                 "properties": {
1464                                     "units": "string",
1465                                     "enum": ["C"]
1466                                 },
1467                                 "x-from-ocf": [
1468                                     "targetvalue = temperature"
1469                                 ]
1470                             },
1471                             {
1472                                 "properties": {
1473                                     "units": "string",
1474                                     "enum": ["F"]
1475                                 },
1476                                 "x-from-ocf": [
1477                                     "targetvalue = (temperature-32)*5/9"
1478                                 ]
1479                             },
1480                             {
1481                                 "properties": {
1482                                     "units": "string",
1483                                     "enum": ["K"]
1484                                 },
1485                                 "x-from-ocf": [
1486                                     "targetvalue = temperature-273.15"
1487                                 ]
1488                             }
1489                         ]
1490                     }
1491                 },
1492                 "minvalue": {
1493                     "type": "number",
1494                     "x-ocf-conversion": {
1495                         "x-ocf-alias": "oic.r.temperature",

```

```

1497         "x-to-ocf": [
1498             "range[0] = minvalue"
1499         ],
1500         "x-from-ocf": [
1501             "minvalue = range[0]",
1502             "otherwise: minvalue = -MAXINT"
1503         ]
1504     }
1505 },
1506     "maxvalue": {
1507         "type": "number",
1508         "x-ocf-conversion": {
1509             "x-ocf-alias": "oic.r.temperature",
1510             "x-to-ocf": [
1511                 "range[1] = maxvalue"
1512             ],
1513             "x-from-ocf": [
1514                 "maxvalue = range[1]",
1515                 "otherwise: maxvalue = MAXINT"
1516             ]
1517         }
1518     },
1519     "step": {
1520         "type": "number",
1521         "x-ocf-conversion": {
1522             "x-ocf-alias": "oic.r.temperature",
1523             "x-to-ocf": [
1524                 "ocf.step = step"
1525             ],
1526             "x-from-ocf": [
1527                 "step = ocf.step",
1528                 "otherwise: step = undefined (0x00)"
1529             ]
1530         }
1531     }
1532 }
1533 },
1534 },
1535     "type": "object",
1536     "allOf": [
1537         { "$ref": "#/definitions/asa.environment.targettemperature" }
1538     ],
1539     "required": [ "targetvalue", "minvalue", "maxvalue", "step" ]
1540 }
1541
1542 responses :
1543 200:
1544     body:
1545         application/json:
1546             schema: /
1547                 {
1548                     "id":
1549 "http://openinterconnect.org/asamapping/schemas/asa.environment.targettemperature.json#",
1550                     "$schema": "http://json-schema.org/draft-04/schema#",
1551                     "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1552 reserved.",
1553                     "title": "Target Temperature",
1554                     "definitions": {
1555                         "asa.environment.targettemperature": {
1556                             "type": "object",
1557                             "properties": {
1558                                 "targetvalue": {
1559                                     "type": "number",
1560                                     "description": "Measured value",
1561                                     "x-ocf-conversion": {
1562                                         "x-ocf-alias": "oic.r.temperature",
1563                                         "x-to-ocf": [
1564                                             "temperature = targetvalue",

```

```

1565         "units = C"
1566     ],
1567     "x-from-ocf": {
1568         "oneOf": [
1569             {
1570                 "properties": {
1571                     "units": "string",
1572                     "enum": ["C"]
1573                 },
1574                 "x-from-ocf": [
1575                     "targetvalue = temperature"
1576                 ]
1577             },
1578             {
1579                 "properties": {
1580                     "units": "string",
1581                     "enum": ["F"]
1582                 },
1583                 "x-from-ocf": [
1584                     "targetvalue = (temperature-32)*5/9"
1585                 ]
1586             },
1587             {
1588                 "properties": {
1589                     "units": "string",
1590                     "enum": ["K"]
1591                 },
1592                 "x-from-ocf": [
1593                     "targetvalue = temperature-273.15"
1594                 ]
1595             }
1596         ]
1597     }
1598 },
1599 "minvalue": {
1600     "type": "number",
1601     "x-ocf-conversion": {
1602         "x-ocf-alias": "oic.r.temperature",
1603         "x-to-ocf": [
1604             "range[0] = minvalue"
1605         ],
1606     },
1607     "x-from-ocf": [
1608         "minvalue = range[0]",
1609         "otherwise: minvalue = -MAXINT"
1610     ]
1611 },
1612 "maxvalue": {
1613     "type": "number",
1614     "x-ocf-conversion": {
1615         "x-ocf-alias": "oic.r.temperature",
1616         "x-to-ocf": [
1617             "range[1] = maxvalue"
1618         ],
1619     },
1620     "x-from-ocf": [
1621         "maxvalue = range[1]",
1622         "otherwise: maxvalue = MAXINT"
1623     ]
1624 },
1625 "step": {
1626     "type": "number",
1627     "x-ocf-conversion": {
1628         "x-ocf-alias": "oic.r.temperature",
1629         "x-to-ocf": [
1630             "ocf.step = step"
1631         ],
1632     },
1633     "x-from-ocf": [
1634         "step = ocf.step",
1635         "otherwise: step = undefined (0x00)"

```

```

1636     ]
1637   }
1638 }
1639 }
1640 },
1641 },
1642 "type": "object",
1643 "allOf": [
1644   { "$ref": "#/definitions/asa.environment.targettemperature" }
1645 ],
1646 "required": [ "targetvalue", "minvalue", "maxvalue", "step" ]
1647 }
1648

```

1649 8.6.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
targetvalue	oic.r.temperature	temperature = targetvalueunits = C	oneOf	측정치
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	

1650 8.6.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/TargetTemperatureResURI		get	post		

1651 8.7 Audio Volume 매핑

1652 8.7.1 개요

1653 이 API 는 OCF Audio Controls 의 인스턴스와 AllJoyn Audio Volume interface 간의 매핑을
1654 정의한다.

1655 8.7.2 URI 예

1656 /AudioVolumeResURI

1657 8.7.3 Resource Type

1658 resource type (rt)는 oic.r.audio 로 정의된다.

1659 8.7.4 RAML 정의

```

1660 #%RAML 0.8
1661 title: AudioVolumeInterfaceMapping
1662 version: OCFv1.0.0-20170317
1663 traits:
1664   - interface-all :

```

```

1665     queryParameters:
1666         if:
1667             enum: ["oic.if.a", "oic.if.baseline"]
1668
1669 /AudioVolumeResURI:
1670     description: |
1671         This API defines the mapping between an instance of an OCF Audio Controls
1672         and the AllJoyn Audio Volume interface.
1673
1674     is : ['interface-all']
1675
1676     get:
1677         responses :
1678             200:
1679                 body:
1680                     application/json:
1681                         schema: /
1682                             {
1683 "http://openinterconnect.org/asamapping/schemas/asa.operation.audiovolume.json#",
1684 "$schema": "http://json-schema.org/draft-04/schema#",
1685 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1686 reserved.",
1687 "title": "Audio Volume",
1688 "definitions": {
1689     "asa.operation.audiovolume": {
1690         "type": "object",
1691         "properties": {
1692             "volume": {
1693                 "type": "integer",
1694                 "description": "Speaker volume index",
1695                 "x-ocf-conversion": {
1696                     "x-ocf-alias": "oic.r.audio",
1697                     "x-to-ocf": [
1698                         "ocf.volume = volume"
1699                     ],
1700                     "x-from-ocf": [
1701                         "volume = ocf.volume"
1702                     ]
1703                 }
1704             },
1705             "maxvolume": {
1706                 "type": "integer",
1707                 "x-ocf-conversion": {
1708                     "x-ocf-alias": "oic.r.audio",
1709                     "x-to-ocf": [
1710                         "range[0] = 0",
1711                         "range[1] = maxvolume"
1712                     ],
1713                     "x-from-ocf": [
1714                         "maxvolume = range[1]",
1715                         "otherwise: maxvalue = 100"
1716                     ]
1717                 }
1718             },
1719             "mute": {
1720                 "type": "boolean",
1721                 "x-ocf-conversion": {
1722                     "x-ocf-alias": "oic.r.audio",
1723                     "x-to-ocf": [
1724                         "ocf.mute = mute"
1725                     ],
1726                     "x-from-ocf": [
1727                         "mute = ocf.mute"
1728                     ]
1729                 }
1730             }
1731         }
1732     }
1733 }

```

```

1730         }
1731     }
1732 }
1733 },
1734 "type": "object",
1735 "allOf": [
1736     { "$ref": "#/definitions/asa.operation.audiovolume" }
1737 ],
1738 "required": [ "volume", "maxvolume", "mute" ]
1739 }
1740
1741 example: /
1742 {
1743     "rt": [ "oic.r.audio" ]
1744 }
1745
1746 post:
1747     body:
1748         application/json:
1749             schema: /
1750                 {
1751                     "id": "http://openinterconnect.org/asamapping/schemas/asa.operation.audiovolume.json#",
1752                     "$schema": "http://json-schema.org/draft-04/schema#",
1753                     "description": "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1754 reserved.",
1755                     "title": "Audio Volume",
1756                     "definitions": {
1757                         "asa.operation.audiovolume": {
1758                             "type": "object",
1759                             "properties": {
1760                                 "volume": {
1761                                     "type": "integer",
1762                                     "description": "Speaker volume index",
1763                                     "x-ocf-conversion": {
1764                                         "x-ocf-alias": "oic.r.audio",
1765                                         "x-to-ocf": [
1766                                             "ocf.volume = volume"
1767                                         ],
1768                                         "x-from-ocf": [
1769                                             "volume = ocf.volume"
1770                                         ]
1771                                     }
1772                                 },
1773                                 "maxvolume": {
1774                                     "type": "integer",
1775                                     "x-ocf-conversion": {
1776                                         "x-ocf-alias": "oic.r.audio",
1777                                         "x-to-ocf": [
1778                                             "range[0] = 0",
1779                                             "range[1] = maxvolume"
1780                                         ],
1781                                         "x-from-ocf": [
1782                                             "maxvolume = range[1]",
1783                                             "otherwise: maxvalue = 100"
1784                                         ]
1785                                     }
1786                                 },
1787                                 "mute": {
1788                                     "type": "boolean",
1789                                     "x-ocf-conversion": {
1790                                         "x-ocf-alias": "oic.r.audio",
1791                                         "x-to-ocf": [
1792                                             "ocf.mute = mute"
1793                                         ],
1794                                         "x-from-ocf": [
1795                                             "mute = ocf.mute"
1796                                         ]
1797                                     }
1798                                 }
1799                             }
1800                         }
1801                     }
1802                 }

```

```

1798         }
1799     }
1800 }
1801 },
1802 "type": "object",
1803 "allOf": [
1804     {"$ref": "#/definitions/asa.operation.audiovolume"}
1805 ],
1806 "required": [ "volume", "maxvolume", "mute" ]
1807 }
1808
1809 responses :
1810     200:
1811         body:
1812             application/json:
1813                 schema: /
1814                     {
1815                         "id":
1816 "http://openinterconnect.org/asamapping/schemas/asa.operation.audiovolume.json#",
1817                         "$schema": "http://json-schema.org/draft-04/schema#",
1818                         "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1819 reserved.",
1820                         "title": "Audio Volume",
1821                         "definitions": {
1822                             "asa.operation.audiovolume": {
1823                                 "type": "object",
1824                                 "properties": {
1825                                     "volume": {
1826                                         "type": "integer",
1827                                         "description": "Speaker volume index",
1828                                         "x-ocf-conversion": {
1829                                             "x-ocf-alias": "oic.r.audio",
1830                                             "x-to-ocf": [
1831                                                 "ocf.volume = volume"
1832                                             ],
1833                                             "x-from-ocf": [
1834                                                 "volume = ocf.volume"
1835                                             ]
1836                                         }
1837                                     },
1838                                     "maxvolume": {
1839                                         "type": "integer",
1840                                         "x-ocf-conversion": {
1841                                             "x-ocf-alias": "oic.r.audio",
1842                                             "x-to-ocf": [
1843                                                 "range[0] = 0",
1844                                                 "range[1] = maxvolume"
1845                                             ],
1846                                             "x-from-ocf": [
1847                                                 "maxvolume = range[1]",
1848                                                 "otherwise: maxvalue = 100"
1849                                             ]
1850                                         }
1851                                     },
1852                                     "mute": {
1853                                         "type": "boolean",
1854                                         "x-ocf-conversion": {
1855                                             "x-ocf-alias": "oic.r.audio",
1856                                             "x-to-ocf": [
1857                                                 "ocf.mute = mute"
1858                                             ],
1859                                             "x-from-ocf": [
1860                                                 "mute = ocf.mute"
1861                                             ]
1862                                         }
1863                                     }
1864                                 }
1865                             }

```



```

1866     },
1867     "type": "object",
1868     "allOf": [
1869       { "$ref": "#/definitions/asa.operation.audiovolume" }
1870     ],
1871     "required": [ "volume", "maxvolume", "mute" ]
1872   }
1873 }

```

1874 8.7.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
volume	oic.r.audio	ocf.volume = volume	volume = ocf.volume	스피커 음량 색인
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	

1875 8.7.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/AudioVolumeResURI		get	post		

1876 8.8 Climate Control Mode 매핑

1877 8.8.1 개요

1878 이 API 는 AllJoyn ClimateControlMode interface 의 인스턴스와 OCF Resource 간의 매핑을
 1879 정의한다. ClimateControlMode 에는 세 가지의 Property 가 있으며, 이들은 다음과 같이 매핑된다:
 1880 mode, supportedmodes -> Mode Resource, operationalstate -> OperationalState Resource.
 1881 이것은 OCF 에서 독립된 두 개의 Resource 인스턴스 또는 두 개의 RT (oic.r.mode,
 1882 oic.r.operationalstate)를 갖는 단일 인스턴스로 표현할 수 있다.

1883 8.8.2 URI 예

1884 /ClimateControlModeResURI

1885 8.8.3 Resource Type

1886 resource type (rt)는 oic.r.mode 로 정의된다.

1887 8.8.4 RAML 정의

```

1888 #%RAML 0.8
1889 title: ClimateControlModeInterfaceMapping
1890 version: OCFv1.0.0-20170317
1891 traits:
1892   - interface-actuator :
1893     queryParameters:
1894       if:
1895         enum: ["oic.if.a", "oic.if.baseline"]
1896
1897 /ClimateControlModeResURI:
1898   description: |
1899     This API defines the mapping between an instance of an AllJoyn ClimateControlMode interface and
1900     the OCF equivalent Resource.
1901     ClimateControlMode has three Properties; these map as follows:

```

```

1902     mode, supportedmodes -> Mode Resource
1903     operationalstate -> OperationalState Resource
1904     This can be represented in OCF either as two distinct Resource instances or a single instance
1905     with two RTs (oic.r.mode, oic.r.operationalstate)
1906
1907     is : ['interface-actuator']
1908
1909     get:
1910         responses :
1911             200:
1912                 body:
1913                     application/json:
1914                         schema: /
1915                             {
1916 "http://openinterconnect.org/asamapping/schemas/asa.operation.climatecontrolmode.json#",
1917 "$schema": "http://json-schema.org/draft-04/schema#",
1918 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
1919 reserved.",
1920 "title": "Climate Control Mode",
1921 "definitions": {
1922     "asa.operation.climatecontrolmode": {
1923         "type": "object",
1924         "properties": {
1925             "mode": {
1926                 "type": "integer",
1927                 "description": "Current mode of device.",
1928                 "x-ocf-conversion": {
1929                     "x-ocf-alias": "oic.r.mode",
1930                     "x-to-ocf": [
1931                         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
1932                         "ocf.mode[0] = modearray[mode]"
1933                     ],
1934                     "x-from-ocf": [
1935                         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
1936                         "mode = indexof modeArray[ocf.mode[0]]"
1937                     ]
1938                 }
1939             },
1940             "supportedmodes": {
1941                 "type": "array",
1942                 "items": {
1943                     "type": "integer"
1944                 },
1945                 "description": "Array of supported modes",
1946                 "x-ocf-conversion": {
1947                     "x-ocf-alias": "oic.r.mode",
1948                     "x-to-ocf": [
1949                         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
1950                         "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
1951 modearray[supportedmodes[x]]"
1952                     ],
1953                     "x-from-ocf": [
1954                         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
1955                         "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
1956 modearray[ocf.supportedmodes[x]]"
1957                     ]
1958                 }
1959             },
1960             "operationalstate": {
1961                 "type": "integer",
1962                 "description": "Current status of device",
1963                 "x-ocf-conversion": {
1964                     "x-ocf-alias": "oic.r.operationalstate",
1965                     "x-to-ocf": [
1966                         "machinestates =
1967 [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
1968                         "currentmachinestate = machinestates[operationalstate]"
1969                     ],

```

```

1970         "x-from-ocf": [
1971             "statearray =
1972 [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
1973             "operationalstate = indexof statearray[currentmachinestate[0]]"
1974         ]
1975     }
1976 }
1977 }
1978 }
1979 },
1980 "type": "object",
1981 "allOf": [
1982     {"$ref": "#/definitions/asa.operation.climatecontrolmode"}
1983 ],
1984 "required": [ "mode","supportedmodes","operationalstate" ]
1985 }
1986
1987 example: /
1988 {
1989     "rt":      ["oic.r.mode","oic.operationalstate"]
1990 }
1991
1992 post:
1993 body:
1994     application/json:
1995
1996     schema: /
1997     {
1998         "id":
1999 "http://openinterconnect.org/asamapping/schemas/asa.operation.climatecontrolmode.json#",
2000         "$schema": "http://json-schema.org/draft-04/schema#",
2001         "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2002 reserved.",
2003         "title": "Climate Control Mode",
2004         "definitions": {
2005             "asa.operation.climatecontrolmode": {
2006                 "type": "object",
2007                 "properties": {
2008                     "mode": {
2009                         "type": "integer",
2010                         "description": "Current mode of device.",
2011                         "x-ocf-conversion": {
2012                             "x-ocf-alias": "oic.r.mode",
2013                             "x-to-ocf": [
2014                                 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2015                                 "ocf.mode[0] = modearray[mode]"
2016                             ],
2017                             "x-from-ocf": [
2018                                 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2019                                 "mode = indexof modeArray[ocf.mode[0]]"
2020                             ]
2021                         },
2022                     "supportedmodes": {
2023                         "type": "array",
2024                         "items": {
2025                             "type": "integer"
2026                         },
2027                         "description": "Array of supported modes",
2028                         "x-ocf-conversion": {
2029                             "x-ocf-alias": "oic.r.mode",
2030                             "x-to-ocf": [
2031                                 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2032                                 "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
2033 modearray[supportedmodes[x]]"
2034                             ],
2035                             "x-from-ocf": [
2036                                 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2037                                 "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof

```

```

2038 modearray[ocf.supportedmodes[x]]"
2039     ]
2040     }
2041   },
2042   "operationalstate": {
2043     "type": "integer",
2044     "description": "Current status of device",
2045     "x-ocf-conversion": {
2046       "x-ocf-alias": "oic.r.operationalstate",
2047       "x-to-ocf": [
2048         "machinestates =
2049 [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
2050         "currentmachinestate = machinestates[operationalstate]"
2051       ],
2052       "x-from-ocf": [
2053         "statearray =
2054 [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
2055         "operationalstate = indexof statearray[currentmachinestate[0]]"
2056       ]
2057     }
2058   }
2059 }
2060 },
2061 },
2062 "type": "object",
2063 "allOf": [
2064   { "$ref": "#/definitions/asa.operation.climatecontrolmode" }
2065 ],
2066 "required": [ "mode","supportedmodes","operationalstate" ]
2067 }
2068
2069 responses :
2070 200:
2071   body:
2072     application/json:
2073       schema: /
2074         {
2075           "id":
2076 "http://openinterconnect.org/asamapping/schemas/asa.operation.climatecontrolmode.json#",
2077           "$schema": "http://json-schema.org/draft-04/schema#",
2078           "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2079 reserved.",
2080           "title": "Climate Control Mode",
2081           "definitions": {
2082             "asa.operation.climatecontrolmode": {
2083               "type": "object",
2084               "properties": {
2085                 "mode": {
2086                   "type": "integer",
2087                   "description": "Current mode of device.",
2088                   "x-ocf-conversion": {
2089                     "x-ocf-alias": "oic.r.mode",
2090                     "x-to-ocf": [
2091                       "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2092                       "ocf.mode[0] = modearray[mode]"
2093                     ],
2094                     "x-from-ocf": [
2095                       "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2096                       "mode = indexof modeArray[ocf.mode[0]]"
2097                     ]
2098                   }
2099               },
2100               "supportedmodes": {
2101                 "type": "array",
2102                 "items": {
2103                   "type": "integer"
2104                 },
2105                 "description": "Array of supported modes",

```

```

2106         "x-ocf-conversion": {
2107             "x-ocf-alias": "oic.r.mode",
2108             "x-to-ocf": [
2109                 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2110                 "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
2111 modearray[supportedmodes[x]]"
2112             ],
2113             "x-from-ocf": [
2114                 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
2115                 "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
2116 modearray[ocf.supportedmodes[x]]"
2117             ]
2118         },
2119     },
2120     "operationalstate": {
2121         "type": "integer",
2122         "description": "Current status of device",
2123         "x-ocf-conversion": {
2124             "x-ocf-alias": "oic.r.operationalstate",
2125             "x-to-ocf": [
2126                 "machinestates =
2127 [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
2128                 "currentmachinestate = machinestates[operationalstate]"
2129             ],
2130             "x-from-ocf": [
2131                 "statearray =
2132 [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
2133                 "operationalstate = indexof statearray[currentmachinestate[0]]"
2134             ]
2135         },
2136     },
2137 },
2138 },
2139 },
2140 "type": "object",
2141 "allOf": [
2142     {"$ref": "#/definitions/asa.operation.climatecontrolmode"}
2143 ],
2144 "required": [ "mode", "supportedmodes", "operationalstate" ]
2145 }
2146

```

2147 8.8.5 Property 정의

['AllJ oyn'] Prope rty name	OCF Resour ce	To OCF	From OCF	설명
suppo rtedm odes	oic.r.mo de	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,Auxilli aryHeat,Dry,ContinuousDr y]for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof modearray[ocf.supported modes[x]]	지원되는 모드의 배열
operat ionalst ate	oic.r.op erationa lstate	machinestates [Idle,Heating,Cooling,PendingHea t,PendingCool,AuxilliaryHeat]curr entmachinestate = machinestates[operationalstate]	statearray [Idle,Heating,Cooling,Pen dingHeat,PendingCool,Au xilliaryHeat]operationalsta te = indexof statearray[currentmachine state[0]]	device 의 현재 상태

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]]	device 의 현재 모드
------	------------	------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------	----------------

8.8.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/ClimateControlModeResURI		get	post		

8.9 Closed Status 매핑

8.9.1 개요

이 API 는 AllJoyn ClosedStatus Interface 의 인스턴스와 OCF Door Resource 간의 매핑을 정의한다.

8.9.2 URI 예

/ClosedStatusResURI

8.9.3 Resource Type

resource type (rt)는 oic.r.door 로 정의된다.

8.9.4 RAML 정의

```

%%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": {

```

```

2190         "isclosed": {
2191             "type": "boolean",
2192             "description": "Open/Closed status Indicator",
2193             "x-ocf-conversion": {
2194                 "x-ocf-alias": "oic.r.door",
2195                 "x-to-ocf": [
2196                     "if isClosed ocf.openState = Closed.",
2197                     "if !isClosed ocf.openState = Open."
2198                 ],
2199                 "x-from-ocf": [
2200                     "isClosed = (openState == Closed)"
2201                 ]
2202             }
2203         },
2204     },
2205     "type": "object",
2206     "allof": [
2207         { "$ref": "#/definitions/asa.operation.closedstatus" }
2208     ],
2209     "required": [ "isclosed" ]
2210 }
2211
2212 example: /
2213 {
2214     "rt":      ["oic.r.door"]
2215 }
2216
2217
2218

```

2219 8.9.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
isclosed	oic.r.door	if isClosed ocf.openState = Closed.if !isClosed ocf.openState = Open.	isClosed = (openState == Closed)	개/폐 상태 지시등

2220 8.9.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/ClosedStatusResURI		get			

2221 8.10 Cycle Control 매핑

2222 8.10.1 개요

2223 이 API 는 AllJoyn CycleControl interface 의 인스턴스와 OCF OperationalState Resource 간의
 2224 매핑을 정의한다. AllJoyn interface 는 Method, ExecuteOperationalCommand 도 지원하며, 이는
 2225 OCF 내에서 oic.r.action collection 내의 oic.r.actuator 의 인스턴스를 사용해서 처리된다. 자세한
 2226 사항은 매핑 스펙의 섹션 8 을 참조하기 바란다.

2227 8.10.2 URI 예

2228 /CycleControlResURI

2229 8.10.3 Resource Type

2230 resource type (rt)는 oic.r.operationalstate 로 정의된다.

2231 8.10.4 RAML 정의

```
2232 #%RAML 0.8
2233 title: CycleControlInterfaceMapping
2234 version: OCFv1.0.0-20170317
2235 traits:
2236   - interface-sensor :
2237     queryParameters:
2238       if:
2239         enum: ["oic.if.s", "oic.if.baseline"]
2240
2241 /CycleControlResURI:
2242   description: |
2243     This API defines the mapping between an instance of an AllJoyn CycleControl interface and the
2244     OCF OperationalState Resource.
2245     The AllJoyn interface also supports a Method, ExecuteOperationalCommand; this is handled in OCF
2246     using an instance of oic.r.actuator within an oic.r.action collection.
2247     Please see Section 8 of the Mapping Specification for specifics.
2248
2249   is : ['interface-sensor']
2250   get:
2251     responses :
2252       200:
2253         body:
2254           application/json:
2255             schema: /
2256               {
2257                 "id":
2258                   "http://openinterconnect.org/asamapping/schemas/asa.operation.ovencyclephase.json#",
2259                 "$schema": "http://json-schema.org/draft-04/schema#",
2260                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2261 reserved.",
2262                 "title": "Oven Cycle Phase",
2263                 "definitions": {
2264                   "asa.operation.ovencyclephase": {
2265                     "type": "object",
2266                     "properties": {
2267                       "cyclephase": {
2268                         "type": "integer",
2269                         "description": "Current phase of the operational cycle",
2270                         "x-ocf-conversion": {
2271                           "x-ocf-alias": "oic.r.operationalstate",
2272                           "x-to-ocf": [
2273                             "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
2274                             "currentmachinestate = phasearray[cyclephase]"
2275                           ],
2276                           "x-from-ocf": [
2277                             "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
2278                             "cyclephase = indexof statearray[currentmachinestate[0]]"
2279                           ]
2280                         }
2281                       },
2282                       "supportedcyclephases": {
2283                         "type": "array",
2284                         "items": {
2285                           "type": "integer"
2286                         },
2287                         "description": "Array of cycle phases supported by the Appliance.",
2288                         "x-ocf-conversion": {
2289                           "x-ocf-alias": "oic.r.operationalstate",
2290                           "x-to-ocf": [
2291                             "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
2292                             "for x=0, x < sizeof(supportedcyclephases): machinestates[x] =
2293 phasearray[supportedcyclephases[x]]"
```



```

2294         },
2295         "x-from-ocf": [
2296             "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
2297             "for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof
2298 phasearray[machinestates[x]]"
2299         ],
2300     },
2301 },
2302 "getvendorphasedescription": {
2303     "x-ocf-type": "method",
2304     "description": "Get cycle phases description",
2305     "x-ocf-conversion": {
2306         "x-ocf-alias": "oic.r.action"
2307     }
2308 },
2309 },
2310 },
2311 },
2312 "type": "object",
2313 "allOf": [
2314     {"$ref": "#/definitions/asa.operation.ovencyclephase"}
2315 ],
2316 "required": [ "cyclephase", "supportedcyclephases" ]
2317 }
2318
2319 example: /
2320 {
2321     "rt":      ["oic.r.operationalstate"]
2322 }
2323

```

2324 8.10.5 Property 정의

['AllJoyn'] Property name	OCF Resour ce	To OCF	From OCF	설명
supportedcy clephases	oic.r.ope rationals tate	phasearray [Unavailable,Preheating,Co oking,Cleaning]for x=0, x < sizeof(supportedcyclephase s): machinestates[x] = phasearray[supportedcyclep hases[x]]	phasearray [Unavailable,Preheating, Cooking,Cleaning]for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof phasearray[machinestate s[x]]	기기에 의해 지원되는 주기 단계의 배열
cyclephase	oic.r.ope rationals tate	phasearray [Unavailable,Preheating,Co oking,Cleaning]currentmachi nestate phasearray[cyclephase]	phasearray [Unavailable,Preheating, Cooking,Cleaning]cycle hase = indexof statearray[currentmachin estate[0]]	동작 주기의 현재 단계
getvendorph asedescripti on	oic.r.acti on			주기 단계 설명 취득

2325 8.10.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/CycleControlResURI		get			

8.11 Fan Speed Level 매핑

8.11.1 개요

이 API 는 AllJoyn FanSpeedLevel interface 의 인스턴스와 OCF AirFlow Resource 간의 매핑을 정의한다. FanSpeedLevel 를 '0x00' (off)로 설정하는 것은 이 interface 에 직접 쓰는 것이 아니라 'OffControl' interface 를 통해 처리된다. 그러한 경우에 Binary Switch 의 인스턴스는 OCF 측에 존재하고, 이는 AirFlowControl 로서 모델링 할 수 있으며, 그리고 나서 Binary Switch 와 AirFlow 의 집합 형태가 된다.

8.11.2 URI 예

/FanSpeedLevelResURI

8.11.3 Resource Type

resource type (rt)는 oic.r.airflow 로 정의된다.

8.11.4 RAML 정의

```
##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",
```

```

2376         "description": "Fan speed level. 0 = off.",
2377         "x-ocf-conversion": {
2378             "x-ocf-alias": "oic.r.airflow",
2379             "x-to-ocf": [
2380                 "speed = fanspeedlevel"
2381             ],
2382             "x-from-ocf": [
2383                 "fanspeedlevel = speed"
2384             ]
2385         },
2386     },
2387     "maxfanspeedlevel": {
2388         "type": "integer",
2389         "description": "Max level allowed for fan speed",
2390         "x-ocf-conversion": {
2391             "x-ocf-alias": "oic.r.airflow",
2392             "x-to-ocf": [
2393                 "range[0] = 0",
2394                 "range[1] = maxfanspeedlevel"
2395             ],
2396             "x-from-ocf": [
2397                 "maxfanspeedlevel = range[1]",
2398                 "otherwise: maxfanspeedlevel = 100"
2399             ]
2400         },
2401     },
2402     "automode": {
2403         "type": "integer",
2404         "description": "Auto mode status.",
2405         "x-ocf-conversion": {
2406             "x-ocf-alias": "oic.r.airflow",
2407             "x-to-ocf": [
2408                 "if automode != NotSupported(0xFF)",
2409                 "  ocf.automode = automode",
2410                 "else no mapping"
2411             ],
2412             "x-from-ocf": [
2413                 "automode = ocf.automode",
2414                 "otherwise: automode = NotSupported(0xFF)"
2415             ]
2416         },
2417     },
2418 },
2419 },
2420 },
2421 "type": "object",
2422 "allOf": [
2423     {"$ref": "#/definitions/asa.operation.fanspeedlevel"}
2424 ],
2425 "required": [ "fanspeedlevel", "maxfanspeedlevel", "automode" ]
2426 }
2427
2428 example: /
2429 {
2430     "rt":      ["oic.r.airflow"]
2431 }
2432
2433 post:
2434 body:
2435 application/json:
2436 schema: /
2437 {
2438     "id":
2439 "http://openinterconnect.org/asamapping/schemas/asa.operation.fanspeedlevel.json#",
2440     "$schema": "http://json-schema.org/draft-04/schema#",
2441     "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2442 reserved.",
2443     "title": "Fan Speed Level",

```

```

2444     "definitions": {
2445         "asa.operation.fanspeedlevel": {
2446             "type": "object",
2447             "properties": {
2448                 "fanspeedlevel": {
2449                     "type": "integer",
2450                     "description": "Fan speed level. 0 = off.",
2451                     "x-ocf-conversion": {
2452                         "x-ocf-alias": "oic.r.airflow",
2453                         "x-to-ocf": [
2454                             "speed = fanspeedlevel"
2455                         ],
2456                         "x-from-ocf": [
2457                             "fanspeedlevel = speed"
2458                         ]
2459                     }
2460                 },
2461                 "maxfanspeedlevel": {
2462                     "type": "integer",
2463                     "description": "Max level allowed for fan speed",
2464                     "x-ocf-conversion": {
2465                         "x-ocf-alias": "oic.r.airflow",
2466                         "x-to-ocf": [
2467                             "range[0] = 0",
2468                             "range[1] = maxfanspeedlevel"
2469                         ],
2470                         "x-from-ocf": [
2471                             "maxfanspeedlevel = range[1]",
2472                             "otherwise: maxfanspeedlevel = 100"
2473                         ]
2474                     }
2475                 },
2476                 "automode": {
2477                     "type": "integer",
2478                     "description": "Auto mode status.",
2479                     "x-ocf-conversion": {
2480                         "x-ocf-alias": "oic.r.airflow",
2481                         "x-to-ocf": [
2482                             "if automode != NotSupported(0xFF)",
2483                             " ocf.automode = automode",
2484                             "else no mapping"
2485                         ],
2486                         "x-from-ocf": [
2487                             "automode = ocf.automode",
2488                             "otherwise: automode = NotSupported(0xFF)"
2489                         ]
2490                     }
2491                 }
2492             }
2493         },
2494         "type": "object",
2495         "allOf": [
2496             { "$ref": "#/definitions/asa.operation.fanspeedlevel" }
2497         ],
2498         "required": [ "fanspeedlevel", "maxfanspeedlevel", "automode" ]
2499     }
2500 }
2501
2502 responses :
2503     200:
2504         body:
2505             application/json:
2506                 schema: /
2507                     {
2508                         "id":
2509 "http://openinterconnect.org/asamapping/schemas/asa.operation.fanspeedlevel.json#",
2510                         "$schema": "http://json-schema.org/draft-04/schema#",
2511                         "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights

```

```

2512 reserved.",
2513 "title": "Fan Speed Level",
2514 "definitions": {
2515     "asa.operation.fanspeedlevel": {
2516         "type": "object",
2517         "properties": {
2518             "fanspeedlevel": {
2519                 "type": "integer",
2520                 "description": "Fan speed level. 0 = off.",
2521                 "x-ocf-conversion": {
2522                     "x-ocf-alias": "oic.r.airflow",
2523                     "x-to-ocf": [
2524                         "speed = fanspeedlevel"
2525                     ],
2526                     "x-from-ocf": [
2527                         "fanspeedlevel = speed"
2528                     ]
2529                 }
2530             },
2531             "maxfanspeedlevel": {
2532                 "type": "integer",
2533                 "description": "Max level allowed for fan speed",
2534                 "x-ocf-conversion": {
2535                     "x-ocf-alias": "oic.r.airflow",
2536                     "x-to-ocf": [
2537                         "range[0] = 0",
2538                         "range[1] = maxfanspeedlevel"
2539                     ],
2540                     "x-from-ocf": [
2541                         "maxfanspeedlevel = range[1]",
2542                         "otherwise: maxfanspeedlevel = 100"
2543                     ]
2544                 }
2545             },
2546             "automode": {
2547                 "type": "integer",
2548                 "description": "Auto mode status.",
2549                 "x-ocf-conversion": {
2550                     "x-ocf-alias": "oic.r.airflow",
2551                     "x-to-ocf": [
2552                         "if automode != NotSupported(0xFF)",
2553                         "  ocf.automode = automode",
2554                         "else no mapping"
2555                     ],
2556                     "x-from-ocf": [
2557                         "automode = ocf.automode",
2558                         "otherwise: automode = NotSupported(0xFF)"
2559                     ]
2560                 }
2561             }
2562         }
2563     },
2564     "type": "object",
2565     "allOf": [
2566         { "$ref": "#/definitions/asa.operation.fanspeedlevel" }
2567     ],
2568     "required": [ "fanspeedlevel", "maxfanspeedlevel", "automode" ]
2569 }
2570
2571

```

2572 8.11.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
maxfanspeedlevel	oic.r.airflow	range[0] = 0range[1] = maxfanspeedlevel	maxfanspeedlevel = range[1]otherwise: maxfanspeedlevel = 100	팬 회전속도에 대해

				허용되는 최고 레벨
automode	oic.r.airflow	if automode != NotSupported(0xFF) ocf.automode = automodeelse no mapping	automode = ocf.automodeotherwise: automode = NotSupported(0xFF)	자동 모드 상태
fanspeedlevel	oic.r.airflow	speed = fanspeedlevel	fanspeedlevel = speed	팬 회전속도 레벨 0 = off

2573 8.11.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/FanSpeedLevelResURI		get	post		

2574 8.12 Heating Zone 매핑

2575 8.12.1 개요

2576 이 API 는 AllJoyn HeatingZone interface 의 인스턴스와 OCF HeatingZoneCollection Resource
 2577 간의 매핑을 정의한다. AllJoyn HeatingZone interface 내의 난방 구역 배열의 각 요소가 자체가 OCF
 2578 HeatingZoneCollection 의 인스턴스의 링크인 OCF HeatingZone 의 인스턴스에 매핑된다. schema
 2579 내에 정의된 매핑은 collection 에 포함된 Resource 를 구성하는 OCF HeatingZone Resource 의
 2580 개체를 설명한다.

2582 8.12.2 URI 예

2583 /HeatingZoneResURI

2584 8.12.3 Resource Type

2585 resource type (rt)는 oic.r.heatingzonecollection 으로 정의된다.

2586 8.12.4 RAML 정의

```

2587 #%RAML 0.8
2588 title: HeatingZoneInterfaceMapping
2589 version: OCFv1.0.0-20170317
2590 traits:
2591   - interface-sensor :
2592     queryParameters:
2593       if:
2594         enum: ["oic.if.s", "oic.if.baseline"]
2595
2596 /HeatingZoneResURI:
2597   description: |
2598     This API defines the mapping between an instance of an AllJoyn HeatingZone interface and an OCF
2599     HeatingZoneCollection Resource.
2600     Each element in the array of heating zones within the AllJoyn HeatingZone interface maps to an
2601     instance of OCF HeatingZone, itself a link in an instance of an OCF HeatingZoneCollection.
2602     The mapping defined in the schema describes the population of the OCF HeatingZone Resource that
2603     constitutes the Resources that are contained in the collection.
2604
2605   is : ['interface-sensor']
  
```

```

2606     get:
2607         responses :
2608             200:
2609                 body:
2610                     application/json:
2611                         schema: /
2612                             {
2613                                 "id":
2614 "http://openinterconnect.org/asamapping/schemas/asa.operation.heatingzone.json#",
2615                                 "$schema": "http://json-schema.org/draft-04/schema#",
2616                                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2617 reserved.",
2618                                 "title": "Heating Zone",
2619                                 "definitions": {
2620                                     "asa.operation.heatingzone": {
2621                                         "type": "object",
2622                                         "properties": {
2623                                             "numberofheatingzones": {
2624                                                 "type": "integer",
2625                                                 "description": "Number of heating zones.",
2626                                                 "x-ocf-conversion": {
2627                                                     "x-ocf-alias": "oic.r.heatingzonecollection",
2628                                                     "x-to-ocf": [
2629                                                         "number of links in the collection = numberofheatingzones"
2630                                                     ],
2631                                                     "x-from-ocf": [
2632                                                         "numberofheatingzones = number of links in the collection"
2633                                                     ]
2634                                                 }
2635                                             },
2636                                             "maxheatinglevels": {
2637                                                 "type": "array",
2638                                                 "items": {
2639                                                     "type": "integer"
2640                                                 },
2641                                                 "description": "Max heating levels for each zone",
2642                                                 "x-ocf-conversion": {
2643                                                     "x-ocf-alias": "oic.r.heatingzone",
2644                                                     "x-to-ocf": [
2645                                                         "Instance of oic.r.heatingzone per array item ",
2646                                                         "for x=0, x<sizeof(maxheatinglevels): ocf.maxheatinglevel =
2647 maxheatinglevels[x]"
2648                                                     ],
2649                                                     "x-from-ocf": [
2650                                                         "for x=0;x<numlinks(oic.r.heatingzonecollection): maxheatinglevels[x] =
2651 ocf.maxheatinglevel"
2652                                                     ]
2653                                                 }
2654                                             },
2655                                             "heatinglevels": {
2656                                                 "type": "array",
2657                                                 "items": {
2658                                                     "type": "integer"
2659                                                 },
2660                                                 "description": "Current heating levels for each zone.",
2661                                                 "x-ocf-conversion": {
2662                                                     "x-ocf-alias": "oic.r.heatingzone",
2663                                                     "x-to-ocf": [
2664                                                         "Instance of oic.r.heatingzone per array item ",
2665                                                         "for x=0, x<sizeof(heatinglevels): ocf.heatinglevel =
2666 maxheatinglevels[x]"
2667                                                     ],
2668                                                     "x-from-ocf": [
2669                                                         "for x=0;x<numlinks(oic.r.heatingzonecollection): heatinglevels[x] =
2670 ocf.heatinglevel"
2671                                                     ]
2672                                                 }
2673                                             }
2674                                         }
2675                             }

```

```

2675     },
2676     },
2677     "type": "object",
2678     "allOf": [
2679         { "$ref": "#/definitions/asa.operation.heatingzone" }
2680     ],
2681     "required": [ "numberOfheatingzones", "maxheatinglevels", "heatinglevels" ]
2682 }
2683
2684 example: /
2685 {
2686     "rt":      [ "oic.r.heatingzonecollection" ]
2687 }
2688

```

2689 8.12.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
heatinglevels	oic.r.heatingzone	Instance of oic.r.heatingzone per array item for x=0, x<sizeof(heatinglevels): ocf.heatinglevel = maxheatinglevels[x]	for x=0;x<numlinks(oic.r.heatingzonecollection): heatinglevels[x] = ocf.heatinglevel	각 구역에 대한 현재 난방 레벨
numberOfheatingzones	oic.r.heatingzonecollection	number of links in the collection = numberOfheatingzones	numberOfheatingzones = number of links in the collection	난방 구역 수
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	각 구역에 대한 최대 난방 레벨

2690 8.12.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/HeatingZoneResURI		get			

2691 8.13 HVAC Fan Mode 매핑

2692 8.13.1 개요

2693 이 API 는 AllJoyn HvacFanMode interface 의 인스턴스와 OCF Mode Resource 간의 매핑을
2694 정의한다.

2695 8.13.2 URI 예

2696 /HvacFanModeResURI

8.13.3 Resource Type

resource type (rt)는 oic.r.mode 로 정의된다.

8.13.4 RAML 정의

```
##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]",

```

```

2757         "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
2758 modearray[supportedmodes[x]]"
2759     ],
2760     "x-from-ocf": [
2761         "modearray = [Auto,Circulation,Continuous]",
2762         "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
2763 modearray[ocf.supportedmodes[x]]"
2764     ]
2765 }
2766 }
2767 }
2768 }
2769 },
2770 "type": "object",
2771 "allOf": [
2772     {"$ref": "#/definitions/asa.operation.hvacfanmode"}
2773 ],
2774 "required": [ "mode", "supportedmodes" ]
2775 }
2776
2777 example: /
2778 {
2779     "rt":      ["oic.r.mode"]
2780 }
2781
2782 post:
2783 body:
2784 application/json:
2785 schema: /
2786 {
2787     "id": "http://openinterconnect.org/asamapping/schemas/asa.operation.hvacfanmode.json#",
2788     "$schema": "http://json-schema.org/draft-04/schema#",
2789     "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2790 reserved.",
2791     "title": "HVAC Fan Mode",
2792     "definitions": {
2793         "asa.operation.hvacfanmode": {
2794             "type": "object",
2795             "properties": {
2796                 "mode": {
2797                     "type": "integer",
2798                     "description": "Current mode of device.",
2799                     "x-ocf-conversion": {
2800                         "x-ocf-alias": "oic.r.mode",
2801                         "x-to-ocf": [
2802                             "modearray = [Auto,Circulation,Continuous]",
2803                             "ocf.mode[0] = modearray[mode]"
2804                         ],
2805                         "x-from-ocf": [
2806                             "modearray = [Auto,Circulation,Continuous]",
2807                             "mode = indexof modeArray[ocf.mode[0]]"
2808                         ]
2809                     }
2810                 },
2811                 "supportedmodes": {
2812                     "type": "array",
2813                     "items": {
2814                         "type": "integer"
2815                     },
2816                     "description": "Array of supported modes",
2817                     "x-ocf-conversion": {
2818                         "x-ocf-alias": "oic.r.mode",
2819                         "x-to-ocf": [
2820                             "modearray = [Auto,Circulation,Continuous]",
2821                             "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
2822 modearray[supportedmodes[x]]"
2823                         ],
2824                         "x-from-ocf": [

```

```

2825         "modearray = [Auto,Circulation,Continuous]",
2826         "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
modearray[ocf.supportedmodes[x]]"
2827     ]
2828 }
2829 }
2830 }
2831 }
2832 }
2833 },
2834 "type": "object",
2835 "allOf": [
2836     {"$ref": "#/definitions/asa.operation.hvacfanmode"}
2837 ],
2838 "required": [ "mode", "supportedmodes" ]
2839 }
2840
2841 responses :
2842 200:
2843     body:
2844         application/json:
2845             schema: /
2846                 {
2847                     "id":
2848 "http://openinterconnect.org/asamapping/schemas/asa.operation.hvacfanmode.json#",
2849                     "$schema": "http://json-schema.org/draft-04/schema#",
2850                     "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
2851                     "title": "HVAC Fan Mode",
2852                     "definitions": {
2853                         "asa.operation.hvacfanmode": {
2854                             "type": "object",
2855                             "properties": {
2856                                 "mode": {
2857                                     "type": "integer",
2858                                     "description": "Current mode of device.",
2859                                     "x-ocf-conversion": {
2860                                         "x-ocf-alias": "oic.r.mode",
2861                                         "x-to-ocf": [
2862                                             "modearray = [Auto,Circulation,Continuous]",
2863                                             "ocf.mode[0] = modearray[mode]"
2864                                         ],
2865                                         "x-from-ocf": [
2866                                             "modearray = [Auto,Circulation,Continuous]",
2867                                             "mode = indexof modeArray[ocf.mode[0]]"
2868                                         ]
2869                                     }
2870                                 },
2871                             },
2872                             "supportedmodes": {
2873                                 "type": "array",
2874                                 "items": {
2875                                     "type": "integer"
2876                                 },
2877                                 "description": "Array of supported modes",
2878                                 "x-ocf-conversion": {
2879                                     "x-ocf-alias": "oic.r.mode",
2880                                     "x-to-ocf": [
2881                                         "modearray = [Auto,Circulation,Continuous]",
2882                                         "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
modearray[supportedmodes[x]]"
2883                                     ],
2884                                     "x-from-ocf": [
2885                                         "modearray = [Auto,Circulation,Continuous]",
2886                                         "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
modearray[ocf.supportedmodes[x]]"
2887                                     ]
2888                                 }
2889                             }
2890                         }
2891                     }
2892                 }

```

```

2893     },
2894     },
2895     "type": "object",
2896     "allOf": [
2897         { "$ref": "#/definitions/asa.operation.hvacfanmode" }
2898     ],
2899     "required": [ "mode", "supportedmodes" ]
2900 }
2901

```

2902 8.13.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
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]]	지원되는 모드 의 배열
mode	oic.r.mode	modearray = [Auto,Circulation,Continuous]ocf.mode[0] = modearray[mode]	modearray = [Auto,Circulation,Continuous]mode = indexof modeArray[ocf.mode[0]]	device 의 현재 모드

2903 8.13.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/HvacFanModeResURI		get	post		

2904 8.14 On Off 매핑

2905 8.14.1 개요

2906 이 API 는 OCF Binary Switch Resource 의 인스턴스와 AllJoyn Interface 집합 간의 매핑을 정의한다.
 2907 검색된 Binary Switch 의 인스턴스가 상시 Operation.OnOffStatus interface 에 매핑된다. Binary
 2908 Switch 의 RETRIEVE 가 Operation.OnOffStatus Interface 의 인스턴스의 동작에 매핑된다. Binary
 2909 Switch 의 UPDATE 가 Operation.OnControl 또는 OffControl 의 메소드 호출에 매핑된다. value =
 2910 true 가 Operation.OnControl 에 매핑되고, value = false 가 Operation.OffControl 에 매핑된다.
 2911

2912 8.14.2 URI 예

2913 /OnOffResURI

2914 8.14.3 Resource Type

2915 resource type (rt)는 oic.r.switch.binary 로 정의된다.

2916 8.14.4 RAML 정의

```

2917 #%RAML 0.8
2918 title: OnOffInterfaceMapping
2919 version: OCFv1.0.0-20170317

```

```

2920 traits:
2921   - interface-actuator :
2922     queryParameters:
2923       if:
2924         enum: ["oic.if.a", "oic.if.baseline"]
2925   - interface-all :
2926     queryParameters:
2927       if:
2928         enum: ["oic.if.s", "oic.if.a", "oic.if.baseline"]
2929
2930 /OnOffResURI:
2931   description: |
2932     This API defines the mapping between an instance of an OCF Binary Switch Resource and the
2933     equivalent Interface set by AllJoyn
2934     A discovered instance of a Binary Switch is always
2935     mapped to an Operation.OnOffStatus interface.
2936     A RETRIEVE on a Binary Switch maps to an action on an instance of an Operation.OnOffStatus
2937     Interface.
2938     An UPDATE on a Binary Switch maps to a method invocation on either Operation.OnControl or
2939     OffControl.
2940     value = true maps to Operation.OnControl
2941     value = false maps to Operation.OffControl
2942
2943   get:
2944     is : ['interface-all']
2945     responses :
2946       200:
2947         body:
2948           application/json:
2949             schema: /
2950               {
2951                 "id":
2952 "http://openinterconnect.org/asamapping/schemas/asa.operation.onoffstatus.json#",
2953                 "$schema": "http://json-schema.org/draft-04/schema#",
2954                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2955 reserved.",
2956                 "title": "On Off Mapping",
2957                 "definitions": {
2958                   "asa.operation.onoffstatus": {
2959                     "type": "object",
2960                     "properties": {
2961                       "onoff": {
2962                         "type": "boolean",
2963                         "description": "On/Off status of the device",
2964                         "x-ocf-conversion": {
2965                           "x-ocf-alias": "oic.r.switch.binary",
2966                           "x-to-ocf": [
2967                             "value = onoff"
2968                           ],
2969                           "x-from-ocf": [
2970                             "onoff = value"
2971                           ]
2972                         }
2973                       }
2974                     }
2975                   },
2976                   "type": "object",
2977                   "allOf": [
2978                     { "$ref": "#/definitions/asa.operation.onoffstatus" }
2979                   ],
2980                   "required": [ "onoff" ]
2981                 }
2982             }
2983

```

```

2984         example: /
2985             {
2986                 "rt":      ["oic.r.switch.binary"]
2987             }
2988
2989     post:
2990         is : ['interface-actuator']
2991     body:
2992         application/json:
2993             schema: /
2994                 {
2995                     "id": "http://openinterconnect.org/asamapping/schemas/asa.operation.oncontrol.json#",
2996                     "$schema": "http://json-schema.org/draft-04/schema#",
2997                     "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
2998 reserved.",
2999                     "title": "On/Off Control",
3000                     "definitions": {
3001                         "oic.r.switch.binary": {
3002                             "properties": {
3003                                 "oneOf": [
3004                                     {
3005                                         "properties": {
3006                                             "value": {
3007                                                 "type": "boolean",
3008                                                 "enum": [true]
3009                                             },
3010                                             "x-ocf-conversion": {
3011                                                 "x-x-from-ocf": [
3012                                                     "asa.operation.oncontrol::switchon()"
3013                                                 ]
3014                                             }
3015                                         }
3016                                     },
3017                                     {
3018                                         "properties": {
3019                                             "value": {
3020                                                 "type": "boolean",
3021                                                 "enum": [false]
3022                                             },
3023                                             "x-ocf-conversion": {
3024                                                 "x-x-from-ocf": [
3025                                                     "asa.operation.offcontrol::switchoff()"
3026                                                 ]
3027                                             }
3028                                         }
3029                                     }
3030                                 ]
3031                             }
3032                         },
3033                         "asa.operation.oncontrol": {
3034                             "type": "object",
3035                             "properties": {
3036                                 "switchon": {
3037                                     "type": "string",
3038                                     "format": "method",
3039                                     "description": "Turn on the device",
3040                                     "x-ocf-conversion": {
3041                                         "x-ocf-alias": "oic.r.switch.binary",
3042                                         "x-to-ocf": [
3043                                             "value = true"
3044                                         ]
3045                                     }
3046                                 }
3047                             }
3048                         },
3049                         "asa.operation.offcontrol": {
3050                             "type": "object",
3051                             "properties": {

```

```

3052         "switchon": {
3053             "type": "string",
3054             "format": "method",
3055             "description": "Turn off the device",
3056             "x-ocf-conversion": {
3057                 "x-ocf-alias": "oic.r.switch.binary",
3058                 "x-to-ocf": [
3059                     "value = false"
3060                 ]
3061             }
3062         }
3063     }
3064 },
3065 },
3066 "type": "object",
3067 "oneOf": [
3068     {"$ref": "#/definitions/oic.r.switch.binary"},
3069     {"$ref": "#/definitions/asa.operation.oncontrol"},
3070     {"$ref": "#/definitions/asa.operation.offcontrol"}
3071 ]
3072 }
3073
3074 responses :
3075 200:
3076     body:
3077         application/json:
3078             schema: /
3079                 {
3080                     "id":
3081 "http://openinterconnect.org/asamapping/schemas/asa.operation.oncontrol.json#",
3082                     "$schema": "http://json-schema.org/draft-04/schema#",
3083                     "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
3084 reserved.",
3085                     "title": "On/Off Control",
3086                     "definitions": {
3087                         "oic.r.switch.binary": {
3088                             "properties": {
3089                                 "oneOf": [
3090                                     {
3091                                         "properties": {
3092                                             "value": {
3093                                                 "type": "boolean",
3094                                                 "enum": [true]
3095                                             },
3096                                             "x-ocf-conversion": {
3097                                                 "x-x-from-ocf": [
3098                                                     "asa.operation.oncontrol::switchon()"
3099                                                 ]
3100                                             }
3101                                         },
3102                                     },
3103                                 ],
3104                                 "properties": {
3105                                     "value": {
3106                                         "type": "boolean",
3107                                         "enum": [false]
3108                                     },
3109                                     "x-ocf-conversion": {
3110                                         "x-x-from-ocf": [
3111                                             "asa.operation.offcontrol::switchoff()"
3112                                         ]
3113                                     }
3114                                 }
3115                             ]
3116                         },
3117                     },
3118                 },
3119                 "asa.operation.oncontrol": {

```

```

3120         "type": "object",
3121         "properties": {
3122             "switchon": {
3123                 "type": "string",
3124                 "format": "method",
3125                 "description": "Turn on the device",
3126                 "x-ocf-conversion": {
3127                     "x-ocf-alias": "oic.r.switch.binary",
3128                     "x-to-ocf": [
3129                         "value = true"
3130                     ]
3131                 }
3132             }
3133         },
3134         "asa.operation.offcontrol": {
3135             "type": "object",
3136             "properties": {
3137                 "switchon": {
3138                     "type": "string",
3139                     "format": "method",
3140                     "description": "Turn off the device",
3141                     "x-ocf-conversion": {
3142                         "x-ocf-alias": "oic.r.switch.binary",
3143                         "x-to-ocf": [
3144                             "value = false"
3145                         ]
3146                     }
3147                 }
3148             }
3149         },
3150     },
3151     "type": "object",
3152     "oneOf": [
3153         {"$ref": "#/definitions/oic.r.switch.binary"},
3154         {"$ref": "#/definitions/asa.operation.oncontrol"},
3155         {"$ref": "#/definitions/asa.operation.offcontrol"}
3156     ]
3157 }
3158 }
3159

```

3160 8.14.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
onoff	oic.r.switch.binary	value = onoff	onoff = value	device 의 On/Off 상태

3161 8.14.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/OnOffResURI		get	post		

3162 8.15 Oven Cycle Phase 매핑

3163 8.15.1 개요

3164 이 API 는 AllJoyn OvenCyclePhase interface 의 인스턴스와 OCF OperationalState Resource 간의
3165 매핑을 정의한다. OvenCyclePhase Property 는 값 0x00-0x7F, 0x80-0xFF 를 사전 정의한다. 여기에
3166 정의된 매핑은 스펙 정의 값에만 해당한다. 모든 제조사 정의 값은 OCF 에서 제조사 정의
3167 Property 를 위한 x.<organization> 구문을 사용해서 표현된다. AllJoyn interface 는 Method,
3168 GetVendorPhasesDescription 도 지원하며, 이는 OCF 에서 oic.r.action collection 내의

3169 oic.r.actuator 의 인스턴스를 사용해서 처리된다. 자세한 사항은 매핑 스펙의 섹션 8 을 참조하기
3170 바란다.

3171 8.15.2 URI 예

3172 /OvenCyclePhaseResURI

3173 8.15.3 Resource Type

3174 resource type (rt)는 oic.r.operationalstate 로 정의된다.

3175 8.15.4 RAML 정의

```
3176 #%RAML 0.8
3177 title: OvenCyclePhaseInterfaceMapping
3178 version: OCFv1.0.0-20170317
3179 traits:
3180   - interface-sensor :
3181       queryParameters:
3182           if:
3183               enum: ["oic.if.s", "oic.if.baseline"]
3184
3185 /OvenCyclePhaseResURI:
3186     description: |
3187         This API defines the mapping between an instance of an AllJoyn OvenCyclePhase interface and the
3188         OCF OperationalState Resource.
3189         OvenCyclePhase cyclephase Property pre-defines values 0x00-0x7F, 0x80-0xFF is for vendor
3190         specific values
3191         The mapping defined herein covers only Spec defined values.
3192         Any vendor defined value shall be represented in OCF using the x.<organization> syntax for a
3193         vendor defined Property.
3194         The AllJoyn interface also supports a Method, GetVendorPhasesDescription; this is handled in
3195         OCF using an instance of oic.r.actuator within an oic.r.action collection.
3196         Please see Section 8 of the Mapping Specification for specifics.
3197
3198     is : ['interface-sensor']
3199     get:
3200         responses :
3201             200:
3202                 body:
3203                     application/json:
3204                         schema: /
3205                             {
3206                                 "id":
3207 "http://openinterconnect.org/asamapping/schemas/asa.operation.ovencyclephase.json#",
3208                                 "$schema": "http://json-schema.org/draft-04/schema#",
3209                                 "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
3210 reserved.",
3211                                 "title": "Oven Cycle Phase",
3212                                 "definitions": {
3213                                     "asa.operation.ovencyclephase": {
3214                                         "type": "object",
3215                                         "properties": {
3216                                             "cyclephase": {
3217                                                 "type": "integer",
3218                                                 "description": "Current phase of the operational cycle",
3219                                                 "x-ocf-conversion": {
3220                                                     "x-ocf-alias": "oic.r.operationalstate",
3221                                                     "x-to-ocf": [
3222                                                         "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
3223                                                         "currentmachinestate = phasearray[cyclephase]"
3224                                                     ],

```

```

3225         "x-from-ocf": [
3226             "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
3227             "cyclephase = indexof statearray[currentmachinestate[0]]"
3228         ]
3229     },
3230 },
3231     "supportedcyclephases": {
3232         "type": "array",
3233         "items": {
3234             "type": "integer"
3235         },
3236         "description": "Array of cycle phases supported by the Appliance.",
3237         "x-ocf-conversion": {
3238             "x-ocf-alias": "oic.r.operationalstate",
3239             "x-to-ocf": [
3240                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
3241                 "for x=0, x < sizeof(supportedcyclephases): machinestates[x] =
3242 phasearray[supportedcyclephases[x]]"
3243             ],
3244             "x-from-ocf": [
3245                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
3246                 "for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof
3247 phasearray[machinestates[x]]"
3248             ]
3249         },
3250     },
3251     "getvendorphasesdescription": {
3252         "x-ocf-type": "method",
3253         "description": "Get cycle phases description",
3254         "x-ocf-conversion": {
3255             "x-ocf-alias": "oic.r.action"
3256         }
3257     }
3258 },
3259 },
3260 },
3261 "type": "object",
3262 "allOf": [
3263     {"$ref": "#/definitions/asa.operation.ovencyclephase"}
3264 ],
3265 "required": [ "cyclephase","supportedcyclephases" ]
3266 }
3267
3268 example: /
3269 {
3270     "rt":      ["oic.r.operationalstate"]
3271 }
3272

```

3273 8.15.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
supportedcy clephases	oic.r.oper ationalsta te	phasearray = [Unavailable,Preheating,Coo king,Cleaning]for x=0, x < sizeof(supportedcyclephase s): machinestates[x] = phasearray[supportedcyclep hases[x]]	phasearray = [Unavailable,Preheating ,Cooking,Cleaning]for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof phasearray[machinestat es[x]]	기기에 의해 지원되는 동작 단계의 배열
cyclephase	oic.r.oper ationalsta te	phasearray = [Unavailable,Preheating,Coo king,Cleaning]currentmachin	phasearray = [Unavailable,Preheating ,Cooking,Cleaning]cycle	현재 동작 단계

		estate phasearray[cyclephase]	=	phase = indexof statearray[currentmachinestate[0]]	
getvendorphasedescription	oic.r.action				동작 단계 설명 취득

3274 **8.15.6 CRUDN 동작**

Resource	Create	Read	Update	Delete	Notify
/OvenCyclePhaseResURI		get			

3275

Annex A Swagger2.0 (단순 정보)

A.1 Audio Volume 매핑

A.1.1 개요

이 API 는 OCF Audio Controls 의 인스턴스와 AllJoyn Audio Volume interface 간의 매핑을 정의한다.

A.1.2 URI 예

/AudioVolumeResURI

A.1.3 Resource Type

resource type (rt)는 ['oic.r.audio']로 정의된다.

A.1.4 Swagger2.0 정의

```
{
  "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\nand the AllJoyn Audio Volume interface.\n",
        "parameters": [
          { "$ref": "#/parameters/interface-all" }
        ],
        "responses": {
          "200": {
            "description": "",
            "x-example": {
              "rt": ["oic.r.audio"]
            }
          }
        }
      }
    }
  }
}
```

```

3330         }
3331     },
3332     "schema": { "$ref": "#/definitions/RetrieveSchema" }
3333 }
3334 },
3335 },
3336 "post": {
3337     "description": "",
3338     "parameters": [
3339         { "$ref": "#/parameters/interface-all" },
3340         {
3341             "name": "body",
3342             "in": "body",
3343             "required": true,
3344             "schema": { "$ref": "#/definitions/UpdateSchema" }
3345         }
3346     ],
3347     "responses": {
3348         "200": {
3349             "description": "",
3350             "schema": { "$ref": "#/definitions/UpdateSchema" }
3351         }
3352     }
3353 }
3354 },
3355 },
3356 "parameters": {
3357     "interface-all" : {
3358         "in" : "query",
3359         "name" : "if",
3360         "type" : "string",
3361         "enum" : ["oic.if.a", "oic.if.baseline"]
3362     }
3363 },
3364 "definitions": {
3365     "RetrieveSchema" :
3366     {
3367         "properties": {
3368             "maxvolume": {
3369                 "type": "integer",
3370                 "x-ocf-conversion": {
3371                     "x-from-ocf": [
3372                         "maxvolume = range[1]",
3373                         "otherwise: maxvalue = 100"
3374                     ],
3375                     "x-ocf-alias": "oic.r.audio",
3376                     "x-to-ocf": [
3377                         "range[0] = 0",
3378                         "range[1] = maxvolume"
3379                     ]
3380                 }
3381             },
3382             "mute": {
3383                 "type": "boolean",
3384                 "x-ocf-conversion": {
3385                     "x-from-ocf": [
3386                         "mute = ocf.mute"
3387                     ],
3388                     "x-ocf-alias": "oic.r.audio",
3389                     "x-to-ocf": [
3390                         "ocf.mute = mute"
3391                     ]
3392                 }
3393             },
3394             "volume": {
3395                 "description": "Speaker volume index",
3396                 "type": "integer",
3397                 "x-ocf-conversion": {
3398                     "x-from-ocf": [
3399                         "volume = ocf.volume"
3400                     ],

```

```

3401         "x-ocf-alias": "oic.r.audio",
3402         "x-to-ocf": [
3403             "ocf.volume = volume"
3404         ]
3405     }
3406 },
3407 },
3408 "required": [
3409     "volume",
3410     "maxvolume",
3411     "mute"
3412 ],
3413 "type": "object"
3414 }
3415
3416 'UpdateSchema' :
3417 {
3418     "properties": {
3419         "maxvolume": {
3420             "type": "integer",
3421             "x-ocf-conversion": {
3422                 "x-from-ocf": [
3423                     "maxvolume = range[1]",
3424                     "otherwise: maxvalue = 100"
3425                 ],
3426                 "x-ocf-alias": "oic.r.audio",
3427                 "x-to-ocf": [
3428                     "range[0] = 0",
3429                     "range[1] = maxvolume"
3430                 ]
3431             }
3432         },
3433         "mute": {
3434             "type": "boolean",
3435             "x-ocf-conversion": {
3436                 "x-from-ocf": [
3437                     "mute = ocf.mute"
3438                 ],
3439                 "x-ocf-alias": "oic.r.audio",
3440                 "x-to-ocf": [
3441                     "ocf.mute = mute"
3442                 ]
3443             }
3444         },
3445         "volume": {
3446             "description": "Speaker volume index",
3447             "type": "integer",
3448             "x-ocf-conversion": {
3449                 "x-from-ocf": [
3450                     "volume = ocf.volume"
3451                 ],
3452                 "x-ocf-alias": "oic.r.audio",
3453                 "x-to-ocf": [
3454                     "ocf.volume = volume"
3455                 ]
3456             }
3457         }
3458     },
3459     "required": [
3460         "volume",
3461         "maxvolume",
3462         "mute"
3463     ],
3464     "type": "object"
3465 }
3466 }
3467 }
3468 }
3469 }
3470

```

3471 A.1.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
volume	oic.r.audio	ocf.volume = volume	volume = ocf.volume	스피커 음량 색인
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	

3472 A.1.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/AudioVolumeResURI		get	post		

3473 A.2 Climate Control Mode 매핑

3474 A.2.1 개요

3475 이 API 는 AllJoyn ClimateControlMode interface 의 인스턴스와 OCF Resource 간의 매핑을
3476 정의한다.

3477 ClimateControlMode 에는 다음과 같이 세 가지 Property 가 있다.

3478 mode, supportedmodes -> Mode Resource

3479 operationalstate -> OperationalState Resource

3480 이것은 OCF 에서 독립된 두 개의 Resource 인스턴스 또는 두 개의 RT(oic.r.mode,
3481 oic.r.operationalstate)를 갖는 단일 인스턴스로 표현할 수 있다.

3482

3483 A.2.2 URI 예

3484 /ClimateControlModeResURI

3485 A.2.3 Resource Type

3486 resource type (rt)는 ['oic.r.mode', 'oic.operationalstate']로 정의된다.

3487 A.2.4 Swagger2.0 정의

```

3488 {
3489   "swagger": "2.0",
3490   "info": {
3491     "title": "Climate Control Mode Mapping",
3492     "version": "OCFv1.0.0-20170317",
3493     "license": {
3494       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
3495       "x-description": "Redistribution and use in source and binary forms, with or without
3496 modification, are permitted provided that the following conditions are met:\n      1.
3497 Redistributions of source code must retain the above copyright notice, this list of conditions and
3498 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
3499 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
3500 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
3501 Connectivity Foundation, INC. \\"AS IS\\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
3502 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
3503 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
3504 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
3505 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
3506 SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND ON
3507 ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
3508 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
  
```

```

3509 OF SUCH DAMAGE.\n"
3510 }
3511 },
3512 "schemes": ["http"],
3513 "consumes": ["application/json"],
3514 "produces": ["application/json"],
3515 "paths": {
3516     "/ClimateControlModeResURI" : {
3517         "get": {
3518             "description": "This API defines the mapping between an instance of an AllJoyn
3519 ClimateControlMode interface and the OCF equivalent Resource.\nClimateControlMode has three
3520 Properties; these map as follows:\n mode, supportedmodes -> Mode Resource\n operationalstate ->
3521 OperationalState Resource\nThis can be represented in OCF either as two distinct Resource instances
3522 or a single instance with two RTs (oic.r.mode, oic.r.operationalstate)\n",
3523             "parameters": [
3524                 {"$ref": "#/parameters/interface-actuator"}
3525             ],
3526             "responses": {
3527                 "200": {
3528                     "description": "",
3529                     "x-example":
3530                     {
3531                         "rt": ["oic.r.mode", "oic.operationalstate"]
3532                     },
3533                     "schema": { "$ref": "#/definitions/RetrieveSchema" }
3534                 }
3535             }
3536         },
3537     },
3538     "post": {
3539         "description": "",
3540         "parameters": [
3541             {"$ref": "#/parameters/interface-actuator"},
3542             {
3543                 "name": "body",
3544                 "in": "body",
3545                 "required": true,
3546                 "schema": { "$ref": "#/definitions/UpdateSchema" }
3547             }
3548         ],
3549         "responses": {
3550             "200": {
3551                 "description": "",
3552                 "schema": { "$ref": "#/definitions/UpdateSchema" }
3553             }
3554         }
3555     }
3556 },
3557 },
3558 "parameters": {
3559     "interface-actuator" : {
3560         "in" : "query",
3561         "name" : "if",
3562         "type" : "string",
3563         "enum" : ["oic.if.a", "oic.if.baseline"]
3564     }
3565 },
3566 "definitions": {
3567     "RetrieveSchema" :
3568     {
3569         "properties": {
3570             "mode": {
3571                 "description": "Current mode of device.",
3572                 "type": "integer",
3573                 "x-ocf-conversion": {
3574                     "x-from-ocf": [
3575                         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3576                         "mode = indexOf modeArray[ocf.mode[0]]"
3577                     ],
3578                     "x-ocf-alias": "oic.r.mode",
3579                     "x-to-ocf": [

```



```

3580         "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3581         "ocf.mode[0] = modearray[mode]"
3582     ]
3583 }
3584 },
3585 "operationalstate": {
3586     "description": "Current status of device",
3587     "type": "integer",
3588     "x-ocf-conversion": {
3589         "x-from-ocf": [
3590             "statearray = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
3591             "operationalstate = indexof statearray[currentmachinestate[0]]"
3592         ],
3593         "x-ocf-alias": "oic.r.operationalstate",
3594         "x-to-ocf": [
3595             "machinestates = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
3596             "currentmachinestate = machinestates[operationalstate]"
3597         ]
3598     }
3599 },
3600 "supportedmodes": {
3601     "description": "Array of supported modes",
3602     "items": {
3603         "type": "integer"
3604     },
3605     "type": "array",
3606     "x-ocf-conversion": {
3607         "x-from-ocf": [
3608             "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3609             "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
3610 modearray[ocf.supportedmodes[x]]"
3611         ],
3612         "x-ocf-alias": "oic.r.mode",
3613         "x-to-ocf": [
3614             "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3615             "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
3616 modearray[supportedmodes[x]]"
3617         ]
3618     }
3619 }
3620 },
3621 "required": [
3622     "mode",
3623     "supportedmodes",
3624     "operationalstate"
3625 ],
3626 "type": "object"
3627 }
3628
3629 ,
3630 "UpdateSchema" :
3631 {
3632     "properties": {
3633         "mode": {
3634             "description": "Current mode of device.",
3635             "type": "integer",
3636             "x-ocf-conversion": {
3637                 "x-from-ocf": [
3638                     "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3639                     "mode = indexof modeArray[ocf.mode[0]]"
3640                 ],
3641                 "x-ocf-alias": "oic.r.mode",
3642                 "x-to-ocf": [
3643                     "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3644                     "ocf.mode[0] = modearray[mode]"
3645                 ]
3646             }
3647         },
3648         "operationalstate": {
3649             "description": "Current status of device",
3650             "type": "integer",

```

```

3651         "x-ocf-conversion": {
3652             "x-from-ocf": [
3653                 "statearray = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
3654                 "operationalstate = indexof statearray[currentmachinestate[0]]"
3655             ],
3656             "x-ocf-alias": "oic.r.operationalstate",
3657             "x-to-ocf": [
3658                 "machinestates = [Idle,Heating,Cooling,PendingHeat,PendingCool,AuxilliaryHeat]",
3659                 "currentmachinestate = machinestates[operationalstate]"
3660             ]
3661         },
3662     },
3663     "supportedmodes": {
3664         "description": "Array of supported modes",
3665         "items": {
3666             "type": "integer"
3667         },
3668         "type": "array",
3669         "x-ocf-conversion": {
3670             "x-from-ocf": [
3671                 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3672                 "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
3673 modearray[ocf.supportedmodes[x]]"
3674             ],
3675             "x-ocf-alias": "oic.r.mode",
3676             "x-to-ocf": [
3677                 "modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat,Dry,ContinuousDry]",
3678                 "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
3679 modearray[supportedmodes[x]]"
3680             ]
3681         }
3682     },
3683 },
3684 "required": [
3685     "mode",
3686     "supportedmodes",
3687     "operationalstate"
3688 ],
3689 "type": "object"
3690 }
3691 }
3692 }
3693 }
3694

```

3695 A.2.5 Property 정의

['AIIJ oyn'] Prope rty name	OCF Resour ce	To OCF	From OCF	설명
suppo rtedm odes	oic.r.mo de	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,Auxilli aryHeat,Dry,ContinuousDr y]for x=0, x < sizeof(supportedmodes): supportedmodes[x] indexof modearray[ocf.supported modes[x]]	지원되는 모드의 배열
operat ionalst ate	oic.r.op erationa lstate	machinestates [Idle,Heating,Cooling,PendingHea t,PendingCool,AuxilliaryHeat]curr entmachinestate machinestates[operationalstate]	statearray [Idle,Heating,Cooling,Pen dingHeat,PendingCool,Au xilliaryHeat]operationalsta te indexof	device 의 현재 상태

			statearray[currentmachine state[0]]	
mode	oic.r.mo de	modearray = [Off,Heat,Cool,Auto,AuxilliaryHeat ,Dry,ContinuousDry]ocf.mode[0] = modearray[mode]	modearray = [Off,Heat,Cool,Auto,Auxilli aryHeat,Dry,ContinuousDr y]mode = indexof modeArray[ocf.mode[0]]	device 의 현재 모드

A.2.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/ClimateControlModeResURI		get	post		

A.3 Closed Status 매핑

A.3.1 개요

이 API 는 AllJoyn ClosedStatus Interface 의 인스턴스와 OCF Door Resource 간의 매핑을 정의한다.

A.3.2 URI 예

/ClosedStatusResURI

A.3.3 Resource Type

resource type (rt)는 ['oic.r.door']로 정의된다.

A.3.4 Swagger2.0 정의

```
{
  "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": [
```

```

3740     {"$ref": "#/parameters/interface-all"}
3741   ],
3742   "responses": {
3743     "200": {
3744       "description": "",
3745       "x-example":
3746         {
3747           "rt": ["oic.r.door"]
3748         },
3749       "schema": { "$ref": "#/definitions/RetrieveSchema" }
3750     }
3751   }
3752 },
3753 },
3754 },
3755 },
3756 "parameters": {
3757   "interface-all" : {
3758     "in" : "query",
3759     "name" : "if",
3760     "type" : "string",
3761     "enum" : ["oic.if.s", "oic.if.baseline"]
3762   }
3763 },
3764 "definitions": {
3765   "RetrieveSchema" :
3766     {
3767       "properties": {
3768         "isclosed": {
3769           "description": "Open/Closed status Indicator",
3770           "type": "boolean",
3771           "x-ocf-conversion": {
3772             "x-from-ocf": [
3773               "isClosed = (openState == Closed)"
3774             ],
3775             "x-ocf-alias": "oic.r.door",
3776             "x-to-ocf": [
3777               "if isClosed ocf.openState = Closed.",
3778               "if !isClosed ocf.openState = Open."
3779             ]
3780           }
3781         }
3782       },
3783       "required": [
3784         "isclosed"
3785       ],
3786       "type": "object"
3787     }
3788 },
3789 },
3790 },
3791

```

A.3.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
isclosed	oic.r.door	if isClosed ocf.openState = Closed.if !isClosed ocf.openState = Open.	isClosed = (openState == Closed)	개/폐 상태 지시등

A.3.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/ClosedStatusResURI		get			

A.4 Air Quality 매핑

A.4.1 개요

이 API 는 AllJoyn AirQuality interface 와 OCF AirQuality Resource 간의 매핑을 정의한다.
복수의 AirQuality interface 의 인스턴스가 있으면 각각의 인스턴스가 OCF AirQuality Resource 의 인스턴스에 매핑된다.
schema 내에 정의된 매핑은 OCF AirQuality Resource 의 개체를 설명한다.
OCF AirQuality Resource 의 단일 인스턴스만 있더라도 OCF AirQualityCollection 의 인스턴스 내에 포함되어야 한다.
collection 내의 링크 수는 노출되는 AllJoyn CurrentAirQuality interface 의 인스턴스의 수와 같다.
OCF 로부터의 매핑 시에 Resource 의 valueType 는 introspected 로 되며, 이 API 는 'Measured'로 설정되었을 때만 적용된다.

A.4.2 URI 예

/CurrentAirQualityResURI

A.4.3 Resource Type

resource type (rt)는 ['oic.r.airqualitycollection']으로 정의된다.

A.4.4 Swagger2.0 정의

```
{
  "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      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
```

```

3846 schema describes the population of the OCF AirQuality Resource.\nEven if there is only a single
3847 instance of an OCF AirQuality Resource this shall be included in an instance of an OCF
3848 AirQualityCollection.\nThe number of links in the collection equates to the number of instances of
3849 the AllJoyn CurrentAirQuality interface that are exposed.\nWhen mapping from OCF the valueType of
3850 the Resource shall be introspected, this API is invoked only if this is set to 'Measured'\n",
3851 "parameters": [
3852   {"$ref": "#/parameters/interface-sensor"}
3853 ],
3854 "responses": {
3855   "200": {
3856     "description": "",
3857     "x-example": {
3858       "rt": ["oic.r.airqualitycollection"]
3859     },
3860     "schema": { "$ref": "#/definitions/RetrieveSchema" }
3861   }
3862 },
3863 },
3864 },
3865 },
3866 },
3867 },
3868 "parameters": {
3869   "interface-sensor" : {
3870     "in" : "query",
3871     "name" : "if",
3872     "type" : "string",
3873     "enum" : ["oic.if.s", "oic.if.baseline"]
3874   }
3875 },
3876 "definitions": {
3877   "RetrieveSchema" :
3878   {
3879     "properties": {
3880       "contaminanttype": {
3881         "description": "The contaminant type",
3882         "type": "integer",
3883         "x-ocf-conversion": {
3884           "x-from-ocf": [
3885             "contaminanttype = indexof contaminanttypearray[ocf.contaminanttype]"
3886           ],
3887           "x-ocf-alias": "oic.r.airquality",
3888           "x-to-ocf": [
3889             "valuetype = Measured",
3890             "contaminanttypearray = [CH2O,CO2,CO,PM2_5,PM10,VOC]",
3891             "ocf.contaminanttype = contaminanttypearray[contaminanttype]"
3892           ]
3893         }
3894       },
3895       "currentvalue": {
3896         "type": "number",
3897         "x-ocf-conversion": {
3898           "x-from-ocf": [
3899             "currentvalue = contaminantvalue"
3900           ],
3901           "x-ocf-alias": "oic.r.airquality",
3902           "x-to-ocf": [
3903             "contaminantvalue = currentvalue"
3904           ]
3905         }
3906       },
3907       "maxvalue": {
3908         "type": "number",
3909         "x-ocf-conversion": {
3910           "x-from-ocf": [
3911             "maxvalue = range[1]"
3912           ],
3913           "x-ocf-alias": "oic.r.airquality",
3914           "x-to-ocf": [
3915             "range[1] = maxvalue"
3916           ]
3917         }
3918       }
3919     }
3920   }
3921 }

```

```

3917     }
3918   },
3919   "minvalue": {
3920     "type": "number",
3921     "x-ocf-conversion": {
3922       "x-from-ocf": [
3923         "minvalue = range[0]"
3924       ],
3925       "x-ocf-alias": "oic.r.airquality",
3926       "x-to-ocf": [
3927         "range[0] = minvalue"
3928       ]
3929     }
3930   },
3931   "precision": {
3932     "type": "number",
3933     "x-ocf-conversion": {
3934       "x-from-ocf": [
3935         "precision = ocf.precision"
3936       ],
3937       "x-ocf-alias": "oic.r.airquality",
3938       "x-to-ocf": [
3939         "ocf.precision = precision"
3940       ]
3941     }
3942   },
3943   "updatemintime": {
3944     "type": "integer",
3945     "x-ocf-conversion": {
3946       "x-from-ocf": [
3947         "updatemintime = ocf.minnotifyperiod"
3948       ],
3949       "x-ocf-alias": "oic.r.value.conditional",
3950       "x-to-ocf": [
3951         "ocf.minnotifyperiod = updatemintime"
3952       ]
3953     }
3954   }
3955 },
3956 "required": [
3957   "contaminanttype",
3958   "currentvalue",
3959   "minvalue",
3960   "maxvalue",
3961   "precision",
3962   "updatemintime"
3963 ],
3964 "type": "object"
3965 }
3966
3967 }
3968
3969

```

3970 A.4.5 Property 정의

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

	y			
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]	오염물질 유형

3971 A.4.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/CurrentAirQualityResURI		get			

3972 A.5 Air Quality Level 매핑

3973 A.5.1 개요

3974 이 API 는 AllJoyn AirQualityLevel interface 와 OCF AirQuality Resource 간의 매핑을 정의한다.

3975 AirQualityLevel interface 의 복수의 인스턴스가 있으면 이들 인스턴스는 각각 OCF AirQuality
3976 Resource 의 인스턴스에 매핑된다.

3977 schema 내에 정의된 매핑은 OCF AirQuality Resource 의 개체를 설명한다.

3978 OCF AirQuality Resource 의 단일 인스턴스만 있더라도 OCF AirQualityCollection 의 인스턴스 내에
3979 포함되어야 한다.

3980 collection 내의 링크 수는 노출되는 AllJoyn CurrentAirQuality interface 의 인스턴스의 수와 같다.

3981 OCF 로부터의 매핑 시에 Resource 의 valueType 는 introspected 로 되며, 이 API 는 'Qualitative'로
3982 설정되었을 때만 적용된다.

3983
3984
3985

3986 A.5.2 URI 예

3987 /CurrentAirQualityLevelResURI

3988 A.5.3 Resource Type

3989 resource type (rt)는 ['oic.r.airqualitycollection']으로 정의된다.

3990 A.5.4 Swagger2.0 정의

```

3991 {
3992   "swagger": "2.0",
3993   "info": {
3994     "title": "Air Quality Level Mapping",
3995     "version": "OCFv1.0.0-20170317",
3996     "license": {
3997       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
3998       "x-description": "Redistribution and use in source and binary forms, with or without
3999 modification, are permitted provided that the following conditions are met:\n      1.
4000 Redistributions of source code must retain the above copyright notice, this list of conditions and
4001 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
4002 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
4003 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open

```



```

4004 Connectivity Foundation, INC. \ "AS IS\ " AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
4005 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
4006 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n          IN NO EVENT SHALL THE Open Connectivity
4007 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
4008 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
4009 SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n          HOWEVER CAUSED AND ON
4010 ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
4011 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
4012 OF SUCH DAMAGE.\n"
4013     }
4014 },
4015 "schemes": ["http"],
4016 "consumes": ["application/json"],
4017 "produces": ["application/json"],
4018 "paths": {
4019     "/CurrentAirQualityLevelResURI" : {
4020         "get": {
4021             "description": "This API defines the mapping between the AllJoyn AirQualityLevel interface
4022 and the OCF AirQuality Resource.\nIf more than one instance of the AirQualityLevel interface is
4023 exposed then each instance maps to an instance of the OCF AirQuality Resource.\nThe mapping defined
4024 in the schema describes the population of the OCF AirQuality Resource.\nEven if there is only a
4025 single instance of an OCF AirQuality Resource then this shall be included in an instance of an OCF
4026 AirQualityCollection.\nThe number of links in the collection equates to the number of instances of
4027 the AllJoyn CurrentAirQuality interface that are exposed.\nWhen mapping from OCF the valueType of
4028 the Resource shall be introspected, this API is invoked only if this is set to 'Qualitative'\n",
4029             "parameters": [
4030                 { "$ref": "#/parameters/interface-sensor" }
4031             ],
4032             "responses": {
4033                 "200": {
4034                     "description": "",
4035                     "x-example": {
4036                         {
4037                             "rt": ["oic.r.airqualitycollection"]
4038                         }
4039                     },
4040                     "schema": { "$ref": "#/definitions/RetrieveSchema" }
4041                 }
4042             }
4043         }
4044     }
4045 },
4046 "parameters": {
4047     "interface-sensor" : {
4048         "in" : "query",
4049         "name" : "if",
4050         "type" : "string",
4051         "enum" : ["oic.if.s", "oic.if.baseline"]
4052     }
4053 },
4054 "definitions": {
4055     "RetrieveSchema" :
4056     {
4057         "properties": {
4058             "contaminanttype": {
4059                 "description": "The contaminant type",
4060                 "type": "integer",
4061                 "x-ocf-conversion": {
4062                     "x-from-ocf": [
4063                         "if ocf.contaminanttype = CH20, contaminanttype = 0",
4064                         "if ocf.contaminanttype = CO2, contaminanttype = 1",
4065                         "if ocf.contaminanttype = CO, contaminanttype = 2",
4066                         "if ocf.contaminanttype = PM2_5, contaminanttype = 3",
4067                         "if ocf.contaminanttype = PM10, contaminanttype = 4",
4068                         "if ocf.contaminanttype = VOC, contaminanttype = 5",
4069                         "if ocf.contaminanttype = Smoke, contaminanttype = 253",
4070                         "if ocf.contaminanttype = Odor, contaminanttype = 254",
4071                         "if ocf.contaminanttype = AirPollution, contaminanttype = 255"
4072                     ],
4073                     "x-ocf-alias": "oic.r.airquality",
4074                     "x-to-ocf": [

```

```

4075         "valuetype = Qualitative",
4076         "if contaminanttype = 0, ocf.contaminanttype = CH2O",
4077         "if contaminanttype = 1, ocf.contaminanttype = CO2",
4078         "if contaminanttype = 2, ocf.contaminanttype = CO",
4079         "if contaminanttype = 3, ocf.contaminanttype = PM2_5",
4080         "if contaminanttype = 4, ocf.contaminanttype = PM10",
4081         "if contaminanttype = 5, ocf.contaminanttype = VOC",
4082         "if contaminanttype = 253, ocf.contaminanttype = Smoke",
4083         "if contaminanttype = 254, ocf.contaminanttype = Odor",
4084         "if contaminanttype = 255, ocf.contaminanttype = AirPollution"
4085     ]
4086 }
4087 },
4088 "currentlevel": {
4089     "type": "integer",
4090     "x-ocf-conversion": {
4091         "x-from-ocf": [
4092             "currentlevel = contaminantvalue"
4093         ],
4094         "x-ocf-alias": "oic.r.airquality",
4095         "x-to-ocf": [
4096             "contaminantvalue = currentlevel"
4097         ]
4098     }
4099 },
4100 "maxlevel": {
4101     "type": "integer",
4102     "x-ocf-conversion": {
4103         "x-from-ocf": [
4104             "maxvalue = range[1]"
4105         ],
4106         "x-ocf-alias": "oic.r.airquality",
4107         "x-to-ocf": [
4108             "range[0] = 0",
4109             "range[1] = maxvalue"
4110         ]
4111     }
4112 },
4113 },
4114 "required": [
4115     "contaminanttype",
4116     "currentlevel",
4117     "maxlevel"
4118 ],
4119 "type": "object"
4120 }
4121 }
4122 }
4123 }
4124

```

4125 A.5.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
currentlevel	oic.r.airquality	contaminantvalue = currentlevel	currentlevel = contaminantvalue	
maxlevel	oic.r.airquality	range[0] = range[1] = maxvalue	maxvalue = range[1]	
contaminanttype	oic.r.airquality	valuetype = Qualitativeif contaminanttype = 0, ocf.contaminanttype = CH2Oif contaminanttype =	if ocf.contaminanttype = CH2O, contaminanttype = 0if ocf.contaminanttype = CO2,	오염 유형 물질

		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	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	
--	--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

4126 A.5.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/CurrentAirQualityLevelResURI		get			

4127 A.6 Current Humidity 매핑

4128 A.6.1 개요

4129 이 API 는 Sensor interface 에 있는 OCF Humidity 의 인스턴스와 AllJoyn Current Humidity
4130 interface 간의 매핑을 정의한다.

4131 습도 센서 상의 RETRIEVE 가 Environment.CurrentHumidity Interface 의 인스턴스 상의 동작에
4132 매핑된다.

4133

4134

4135 A.6.2 URI 예

4136 /CurrentHumidityResURI

4137 A.6.3 Resource Type

4138 resource type (rt)는 ['oic.r.humidity']로 정의된다.

4139 A.6.4 Swagger2.0 정의

```
4140 {
4141   "swagger": "2.0",
4142   "info": {
4143     "title": "Current Humidity Mapping",
4144     "version": "OCFv1.0.0-20170317",
4145     "license": {
4146       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
4147       "x-description": "Redistribution and use in source and binary forms, with or without
4148 modification, are permitted provided that the following conditions are met:\n      1.
4149 Redistributions of source code must retain the above copyright notice, this list of conditions and
4150 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
4151 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
4152 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
4153 Connectivity Foundation, INC. \\"AS IS\\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
4154 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
4155 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
4156 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
4157 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
4158 SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND ON
4159 ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
4160 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
4161 OF SUCH DAMAGE.\n"
4162     }
4163   },
4164   "schemes": ["http"],
4165   "consumes": ["application/json"],
4166   "produces": ["application/json"],
4167   "paths": {
4168     "/CurrentHumidityResURI" : {
4169       "get": {
4170         "description": "This API defines the mapping between an instance of an OCF Humidity which
4171 exposes only a sensor interface\nand the AllJoyn Current Humidity interface.\nA RETRIEVE on a
4172 Temperature Sensor maps to an action on an instance of an Environment.CurrentTemperature
4173 Interface.\n",
4174         "parameters": [
4175           { "$ref": "#/parameters/interface-sensor" }
4176         ],
4177         "responses": {
4178           "200": {
4179             "description": "",
4180             "x-example": {
4181               "rt": ["oic.r.humidity"]
4182             },
4183             "schema": { "$ref": "#/definitions/RetrieveSchema" }
4184           }
4185         }
4186       }
4187     }
4188   },
4189   "parameters": {
4190     "interface-sensor" : {
4191       "in" : "query",
4192       "name" : "if",
4193       "type" : "string",
4194       "enum" : ["oic.if.s", "oic.if.baseline"]
4195     }
4196   },
4197   "definitions": {
4198     "RetrieveSchema" :
4199     {
4200       "properties": {
4201         "currentvalue": {
4202           "description": "Measured value",
4203           "type": "number",
4204           "x-ocf-conversion": {
4205             "x-from-ocf": [
4206               "currentvalue = humidity"
4207             ]
4208           }
4209         }
4210       }
4211     }
4212   }
4213 }
```

```

4209         ],
4210         "x-ocf-alias": "oic.r.humidity",
4211         "x-to-ocf": [
4212             "humidity = currentValue"
4213         ]
4214     },
4215 },
4216 "maxvalue": {
4217     "description": "Max measured value for humidity",
4218     "type": "number",
4219     "x-ocf-conversion": {
4220         "x-from-ocf": [
4221             "maxvalue = range[1]"
4222         ],
4223         "x-ocf-alias": "oic.r.humidity",
4224         "x-to-ocf": [
4225             "range[0] = 0",
4226             "range[1] = maxvalue"
4227         ]
4228     }
4229 },
4230 },
4231 "required": [
4232     "currentvalue",
4233     "maxvalue"
4234 ],
4235 "type": "object"
4236 }
4237 }
4238 }
4239 }
4240

```

4241 A.6.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
currentvalue	oic.r.humidity	humidity currentValue	= currentvalue = humidity	측정치
maxvalue	oic.r.humidity	range[0] Orange[1] maxvalue	= maxvalue = range[1]	습도에 대한 최대 측정치

4242 A.6.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/CurrentHumidityResURI		get			

4243 A.7 Current Temperature 매핑

4244 A.7.1 개요

4245 이 API 는 Sensor interface 에 있는 OCF Temperature 의 인스턴스와 AllJoyn Current Temperature
4246 interface 간의 매핑을 정의한다.

4247 온도 센서 상의 RETRIEVE 가 Environment.CurrentTemperature Interface 의 인스턴스 상의 동작에
4248 매핑된다.

4251 A.7.2 URI 예

4252 /CurrentTemperatureResURI

A.7.3 Resource Type

resource type (rt)는 ['oic.r.temperature']로 정의된다.

A.7.4 Swagger2.0 정의

```
{
  "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\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\and 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": {
```

```

4320     "description": "Measured value",
4321     "type": "number",
4322     "x-ocf-conversion": {
4323         "x-from-ocf": {
4324             "oneOf": [
4325                 {
4326                     "properties": {
4327                         "enum": [
4328                             "C"
4329                         ],
4330                         "units": "string"
4331                     },
4332                     "x-from-ocf": [
4333                         "currentvalue = temperature"
4334                     ]
4335                 },
4336                 {
4337                     "properties": {
4338                         "enum": [
4339                             "F"
4340                         ],
4341                         "units": "string"
4342                     },
4343                     "x-from-ocf": [
4344                         "currentvalue = (temperature-32)*5/9"
4345                     ]
4346                 },
4347                 {
4348                     "properties": {
4349                         "enum": [
4350                             "K"
4351                         ],
4352                         "units": "string"
4353                     },
4354                     "x-from-ocf": [
4355                         "currentvalue = temperature-273.15"
4356                     ]
4357                 }
4358             ]
4359         },
4360         "x-ocf-alias": "oic.r.temperature",
4361         "x-to-ocf": [
4362             "temperature = currentValue",
4363             "units = C"
4364         ]
4365     },
4366     "precision": {
4367         "type": "number",
4368         "x-ocf-conversion": {
4369             "x-from-ocf": [
4370                 "precision = ocf.precision"
4371             ],
4372             "x-ocf-alias": "oic.r.temperature",
4373             "x-to-ocf": [
4374                 "ocf.precision = precision"
4375             ]
4376         }
4377     },
4378     "updatemintime": {
4379         "type": "integer",
4380         "x-ocf-conversion": {
4381             "x-from-ocf": [
4382                 "updatemintime = ocf.minnotifyperiod"
4383             ],
4384             "x-ocf-alias": "oic.r.value.conditional",
4385             "x-to-ocf": [
4386                 "ocf.minnotifyperiod = updatemintime"
4387             ]
4388         }
4389     }
4390 }

```

```

4391     },
4392     "required": [
4393         "currentvalue",
4394         "precision",
4395         "updatemintime"
4396     ],
4397     "type": "object"
4398 }
4399
4400 }
4401 }
4402

```

4403 A.7.5 Property 정의

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

4404 A.7.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/CurrentTemperatureResURI		get			

4405 A.8 Cycle Control 매핑

4406 A.8.1 개요

4407 이 API 는 AllJoyn CycleControl interface 의 인스턴스와 OCF OperationalState Resource 간의
4408 매핑을 정의한다.

4409 AllJoyn interface 는 Method, ExecuteOperationalCommand 도 지원하면, 이는 OCF 내에서
4410 oic.r.action collection 내의 oic.r.actuator 의 인스턴스를 사용해서 처리된다.
4411 자세한 사항은 매핑 스펙의 섹션 8 을 참조하기 바란다.
4412

4413 A.8.2 URI 예

4414 /CycleControlResURI

4415 A.8.3 Resource Type

4416 resource type (rt)는 ['oic.r.operationalstate']로 정의된다.

4417 A.8.4 Swagger2.0 정의

```

4418 {
4419     "swagger": "2.0",
4420     "info": {
4421         "title": "Cycle Control Mapping",
4422         "version": "OCFv1.0.0-20170317",
4423         "license": {
4424             "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
4425             "x-description": "Redistribution and use in source and binary forms, with or without

```



```

4426 modification, are permitted provided that the following conditions are met:\n      1.
4427 Redistributions of source code must retain the above copyright notice, this list of conditions and
4428 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
4429 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
4430 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
4431 Connectivity Foundation, INC. \n"AS IS\n" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
4432 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
4433 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
4434 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
4435 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
4436 SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND ON
4437 ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
4438 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
4439 OF SUCH DAMAGE.\n"
4440 }
4441 },
4442 "schemes": ["http"],
4443 "consumes": ["application/json"],
4444 "produces": ["application/json"],
4445 "paths": {
4446   "/CycleControlResURI" : {
4447     "get": {
4448       "description": "This API defines the mapping between an instance of an AllJoyn CycleControl
4449 interface and the OCF OperationalState Resource.\nThe AllJoyn interface also supports a Method,
4450 ExecuteOperationalCommand; this is handled in OCF using an instance of oic.r.actuator within an
4451 oic.r.action collection.\nPlease see Section 8 of the Mapping Specification for specifics.\n",
4452       "parameters": [
4453         {"$ref": "#/parameters/interface-sensor"}
4454       ],
4455       "responses": {
4456         "200": {
4457           "description": "",
4458           "x-example": {
4459             "rt": ["oic.r.operationalstate"]
4460           },
4461           "schema": { "$ref": "#/definitions/RetrieveSchema" }
4462         }
4463       }
4464     }
4465   }
4466 },
4467 },
4468 },
4469 "parameters": {
4470   "interface-sensor" : {
4471     "in" : "query",
4472     "name" : "if",
4473     "type" : "string",
4474     "enum" : ["oic.if.s", "oic.if.baseline"]
4475   }
4476 },
4477 "definitions": {
4478   "RetrieveSchema" :
4479   {
4480     "properties": {
4481       "cyclephase": {
4482         "description": "Current phase of the operational cycle",
4483         "type": "integer",
4484         "x-ocf-conversion": {
4485           "x-from-ocf": [
4486             "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
4487             "cyclephase = indexof statearray[currentmachinestate[0]]"
4488           ],
4489           "x-ocf-alias": "oic.r.operationalstate",
4490           "x-to-ocf": [
4491             "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
4492             "currentmachinestate = phasearray[cyclephase]"
4493           ]
4494         }
4495       },
4496       "getvendorphasedescription": {

```

```

4497         "description": "Get cycle phases description",
4498         "x-ocf-conversion": {
4499             "x-ocf-alias": "oic.r.action"
4500         },
4501         "x-ocf-type": "method"
4502     },
4503     "supportedcyclephases": {
4504         "description": "Array of cycle phases supported by the Appliance.",
4505         "items": {
4506             "type": "integer"
4507         },
4508         "type": "array",
4509         "x-ocf-conversion": {
4510             "x-from-ocf": [
4511                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
4512                 "for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof
4513 phasearray[machinestates[x]]"
4514             ],
4515             "x-ocf-alias": "oic.r.operationalstate",
4516             "x-to-ocf": [
4517                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
4518                 "for x=0, x < sizeof(supportedcyclephases): machinestates[x] =
4519 phasearray[supportedcyclephases[x]]"
4520             ]
4521         }
4522     },
4523     },
4524     "required": [
4525         "cyclephase",
4526         "supportedcyclephases"
4527     ],
4528     "type": "object"
4529 }
4530 }
4531 }
4532 }
4533 }

```

4534 A.8.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
supported cyclephas es	oic.r.opera tionalstate	phasearray = [Unavailable,Preheating,Cook ing,Cleaning]for x=0, x < sizeof(supportedcyclephases) : machinestates[x] = phasearray[supportedcycleph ases[x]]	phasearray = [Unavailable,Preheating ,Cooking,Cleaning]for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof phasearray[machinestat es[x]]	기기에 의해 지원되는 주기 위상의 배열
cyclephas e	oic.r.opera tionalstate	phasearray = [Unavailable,Preheating,Cook ing,Cleaning]currentmachines tate = phasearray[cyclephase]	phasearray = [Unavailable,Preheating ,Cooking,Cleaning]cycle phase = indexof statearray[currentmachi nestate[0]]	동작 주기의 현재 위상
getvendor phasedesc ription	oic.r.actio n			주기 위상 설명 취득

4535 A.8.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
----------	--------	------	--------	--------	--------

/CycleControlResURI		get			
---------------------	--	-----	--	--	--

A.9 Fan Speed Level 매핑

A.9.1 개요

이 API 는 AllJoyn FanSpeedLevel interface 의 인스턴스와 OCF AirFlow Resource 간의 매핑을 정의한다.

FanSpeedLevel 를 '0x00' (off)로 설정하는 것은 이 interface 에 직접 쓰는 것이 아니라 'OffControl' interface 를 통해 처리된다.

그러한 경우에 Binary Switch 의 인스턴스가 OCF 측에 있고, 이는 AirFlowControl 로서 모델링 할 수 있으며, 그리고 나서 Binary Switch 와 AirFlow 의 collection 으로 정의된다.

A.9.2 URI 예

/FanSpeedLevelResURI

A.9.3 Resource Type

resource type (rt)는 ['oic.r.airflow']로 정의된다.

A.9.4 Swagger2.0 정의

```
{
  "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": {
```

```

4589         "200": {
4590             "description": "",
4591             "x-example":
4592                 {
4593                     "rt": ["oic.r.airflow"]
4594                 }
4595             ,
4596             "schema": { "$ref": "#/definitions/RetrieveSchema" }
4597         }
4598     },
4599 },
4600 "post": {
4601     "description": "",
4602     "parameters": [
4603         { "$ref": "#/parameters/interface-actuator" },
4604         {
4605             "name": "body",
4606             "in": "body",
4607             "required": true,
4608             "schema": { "$ref": "#/definitions/UpdateSchema" }
4609         }
4610     ],
4611     "responses": {
4612         "200": {
4613             "description": "",
4614             "schema": { "$ref": "#/definitions/UpdateSchema" }
4615         }
4616     }
4617 },
4618 },
4619 },
4620 "parameters": {
4621     "interface-actuator" : {
4622         "in" : "query",
4623         "name" : "if",
4624         "type" : "string",
4625         "enum" : ["oic.if.a", "oic.if.baseline"]
4626     }
4627 },
4628 "definitions": {
4629     "RetrieveSchema" :
4630     {
4631         "properties": {
4632             "automode": {
4633                 "description": "Auto mode status.",
4634                 "type": "integer",
4635                 "x-ocf-conversion": {
4636                     "x-from-ocf": [
4637                         "automode = ocf.automode",
4638                         "otherwise: automode = NotSupported(0xFF)"
4639                     ],
4640                     "x-ocf-alias": "oic.r.airflow",
4641                     "x-to-ocf": [
4642                         "if automode != NotSupported(0xFF)",
4643                         "ocf.automode = automode",
4644                         "else no mapping"
4645                     ]
4646                 }
4647             },
4648             "fanspeedlevel": {
4649                 "description": "Fan speed level. 0 = off.",
4650                 "type": "integer",
4651                 "x-ocf-conversion": {
4652                     "x-from-ocf": [
4653                         "fanspeedlevel = speed"
4654                     ],
4655                     "x-ocf-alias": "oic.r.airflow",
4656                     "x-to-ocf": [
4657                         "speed = fanspeedlevel"
4658                     ]
4659                 }
4660             }
4661         }
4662     }
4663 }

```

```

4660     },
4661     "maxfanspeedlevel": {
4662         "description": "Max level allowed for fan speed",
4663         "type": "integer",
4664         "x-ocf-conversion": {
4665             "x-from-ocf": [
4666                 "maxfanspeedlevel = range[1]",
4667                 "otherwise: maxfanspeedlevel = 100"
4668             ],
4669             "x-ocf-alias": "oic.r.airflow",
4670             "x-to-ocf": [
4671                 "range[0] = 0",
4672                 "range[1] = maxfanspeedlevel"
4673             ]
4674         }
4675     },
4676 },
4677 "required": [
4678     "fanspeedlevel",
4679     "maxfanspeedlevel",
4680     "automode"
4681 ],
4682 "type": "object"
4683 }
4684
4685 ,
4686 "UpdateSchema" :
4687 {
4688     "properties": {
4689         "automode": {
4690             "description": "Auto mode status.",
4691             "type": "integer",
4692             "x-ocf-conversion": {
4693                 "x-from-ocf": [
4694                     "automode = ocf.automode",
4695                     "otherwise: automode = NotSupported(0xFF)"
4696                 ],
4697                 "x-ocf-alias": "oic.r.airflow",
4698                 "x-to-ocf": [
4699                     "if automode != NotSupported(0xFF)",
4700                     " ocf.automode = automode",
4701                     "else no mapping"
4702                 ]
4703             }
4704         },
4705         "fanspeedlevel": {
4706             "description": "Fan speed level. 0 = off.",
4707             "type": "integer",
4708             "x-ocf-conversion": {
4709                 "x-from-ocf": [
4710                     "fanspeedlevel = speed"
4711                 ],
4712                 "x-ocf-alias": "oic.r.airflow",
4713                 "x-to-ocf": [
4714                     "speed = fanspeedlevel"
4715                 ]
4716             }
4717         },
4718         "maxfanspeedlevel": {
4719             "description": "Max level allowed for fan speed",
4720             "type": "integer",
4721             "x-ocf-conversion": {
4722                 "x-from-ocf": [
4723                     "maxfanspeedlevel = range[1]",
4724                     "otherwise: maxfanspeedlevel = 100"
4725                 ],
4726                 "x-ocf-alias": "oic.r.airflow",
4727                 "x-to-ocf": [
4728                     "range[0] = 0",
4729                     "range[1] = maxfanspeedlevel"
4730                 ]

```

```

4731     }
4732   },
4733 },
4734 "required": [
4735   "fanspeedlevel",
4736   "maxfanspeedlevel",
4737   "automode"
4738 ],
4739 "type": "object"
4740 }
4741 }
4742 }
4743 }
4744

```

4745 A.9.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
maxfanspeedlevel	oic.r.airflow	range[0] = 0range[1] = maxfanspeedlevel	maxfanspeedlevel = range[1]otherwise: maxfanspeedlevel = 100	팬 회전속도에 대해 허용되는 최고 레벨
automode	oic.r.airflow	if automode != NotSupported(0xFF) ocf.automode = automodeelse no mapping	automode = ocf.automodeotherwise: automode = NotSupported(0xFF)	자동 모드 상태
fanspeedlevel	oic.r.airflow	speed = fanspeedlevel	fanspeedlevel = speed	팬 회전속도 레벨 0 = off

4746 A.9.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/FanSpeedLevelResURI		get	post		

4747 A.10 Heating Zone 매핑

4748 A.10.1 개요

4749 이 API 는 AllJoyn HeatingZone interface 의 인스턴스와 OCF HeatingZoneCollection Resource
4750 간의 매핑을 정의한다.

4751 AllJoyn HeatingZone interface 내의 난방 구역 배열의 각 요소가 자체가 OCF

4752 HeatingZoneCollection 의 인스턴스의 링크인 OCF HeatingZone 의 인스턴스에 매핑된다.

4753 schema 내에 정의된 매핑은 collection 에 포함된 Resource 를 구성하는 OCF HeatingZone
4754 Resource 의 개체를 설명한다.

4755

4756 A.10.2 URI 예

4757 /HeatingZoneResURI

A.10.3 Resource Type

resource type (rt)는 ['oic.r.heatingzonecollection']으로 정의된다.

A.10.4 Swagger2.0 정의

```
{
  "swagger": "2.0",
  "info": {
    "title": "Heating Zone 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": {
    "/HeatingZoneResURI" : {
      "get": {
        "description": "This API defines the mapping between an instance of an AllJoyn HeatingZone
interface and an OCF HeatingZoneCollection Resource.\nEach 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.\nThe mapping defined in the schema describes the
population of the OCF HeatingZone Resource that constitutes the Resources that are contained in the
collection.\n",
        "parameters": [
          { "$ref": "#/parameters/interface-sensor" }
        ],
        "responses": {
          "200": {
            "description": "",
            "x-example": {
              "rt": ["oic.r.heatingzonecollection"]
            },
            "schema": { "$ref": "#/definitions/RetrieveSchema" }
          }
        }
      }
    }
  },
  "parameters": {
    "interface-sensor" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.s", "oic.if.baseline"]
    }
  },
  "definitions": {
    "RetrieveSchema" :
  }
```

```

4825     "properties": {
4826         "heatinglevels": {
4827             "description": "Current heating levels for each zone.",
4828             "items": {
4829                 "type": "integer"
4830             },
4831             "type": "array",
4832             "x-ocf-conversion": {
4833                 "x-from-ocf": [
4834                     "for x=0;x<numlinks(oic.r.heatingzonecollection): heatinglevels[x] =
4835 ocf.heatinglevel"
4836                 ],
4837                 "x-ocf-alias": "oic.r.heatingzone",
4838                 "x-to-ocf": [
4839                     "Instance of oic.r.heatingzone per array item ",
4840                     "for x=0, x<sizeof(heatinglevels): ocf.heatinglevel = maxheatinglevels[x]"
4841                 ]
4842             },
4843         },
4844         "maxheatinglevels": {
4845             "description": "Max heating levels for each zone",
4846             "items": {
4847                 "type": "integer"
4848             },
4849             "type": "array",
4850             "x-ocf-conversion": {
4851                 "x-from-ocf": [
4852                     "for x=0;x<numlinks(oic.r.heatingzonecollection): maxheatinglevels[x] =
4853 ocf.maxheatinglevel"
4854                 ],
4855                 "x-ocf-alias": "oic.r.heatingzone",
4856                 "x-to-ocf": [
4857                     "Instance of oic.r.heatingzone per array item ",
4858                     "for x=0, x<sizeof(maxheatinglevels): ocf.maxheatinglevel = maxheatinglevels[x]"
4859                 ]
4860             },
4861         },
4862         "numberofheatingzones": {
4863             "description": "Number of heating zones.",
4864             "type": "integer",
4865             "x-ocf-conversion": {
4866                 "x-from-ocf": [
4867                     "numberofheatingzones = number of links in the collection"
4868                 ],
4869                 "x-ocf-alias": "oic.r.heatingzonecollection",
4870                 "x-to-ocf": [
4871                     "number of links in the collection = numberofheatingzones"
4872                 ]
4873             },
4874         },
4875     },
4876     "required": [
4877         "numberofheatingzones",
4878         "maxheatinglevels",
4879         "heatinglevels"
4880     ],
4881     "type": "object"
4882 }
4883
4884 }
4885 }
4886

```

4887 A.10.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
heatinglevels	oic.r.heatingzone	Instance of oic.r.heatingzone	for x=0;x<numlinks(oic.r.heatingz	각

		per array item for x=0, x<sizeof(heatinglevels): ocf.heatinglevel = maxheatinglevels[x]]	oncollection): heatinglevels[x] ocf.heatinglevel =	구역 에 대한 현재 난방 레벨
numberofheatingzones	oic.r.heatingzone collection	number of links in the collection = numberofheatingzones	numberofheatingzones = number of links in the collection	난방 구역 수
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 =	각 구역 에 대한 최대 난방 레벨

4888 A.10.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/HeatingZoneResURI		get			

4889 A.11 HVAC Fan Mode 매핑

4890 A.11.1 개요

4891 이 API 는 AllJoyn HvacFanMode interface 의 인스턴스와 OCF Mode Resource 간의 매핑을
4892 정의한다.

4894 A.11.2 URI 예

4895 /HvacFanModeResURI

4896 A.11.3 Resource Type

4897 resource type (rt)는 ['oic.r.mode']로 정의된다.

4898 A.11.4 Swagger2.0 정의

```

4899 {
4900   "swagger": "2.0",
4901   "info": {
4902     "title": "HVAC Fan Mode Mapping",
4903     "version": "OCFv1.0.0-20170317",
4904     "license": {
4905       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
4906       "x-description": "Redistribution and use in source and binary forms, with or without
4907 modification, are permitted provided that the following conditions are met:\n      1.
4908 Redistributions of source code must retain the above copyright notice, this list of conditions and
4909 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above

```

```

4910 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
4911 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
4912 Connectivity Foundation, INC. \ "AS IS\ " AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
4913 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
4914 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
4915 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
4916 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
4917 SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND ON
4918 ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
4919 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
4920 OF SUCH DAMAGE.\n"
4921 }
4922 },
4923 "schemes": ["http"],
4924 "consumes": ["application/json"],
4925 "produces": ["application/json"],
4926 "paths": {
4927   "/HvacFanModeResURI" : {
4928     "get": {
4929       "description": "This API defines the mapping between an instance of an AllJoyn HvacFanMode
4930 interface and an OCF Mode Resource.\n",
4931       "parameters": [
4932         { "$ref": "#/parameters/interface-actuator" }
4933       ],
4934       "responses": {
4935         "200": {
4936           "description": "",
4937           "x-example": {
4938             "rt": ["oic.r.mode"]
4939           },
4940           "schema": { "$ref": "#/definitions/RetrieveSchema" }
4941         }
4942       }
4943     },
4944     "post": {
4945       "description": "",
4946       "parameters": [
4947         { "$ref": "#/parameters/interface-actuator" },
4948         {
4949           "name": "body",
4950           "in": "body",
4951           "required": true,
4952           "schema": { "$ref": "#/definitions/UpdateSchema" }
4953         }
4954       ],
4955       "responses": {
4956         "200": {
4957           "description": "",
4958           "schema": { "$ref": "#/definitions/UpdateSchema" }
4959         }
4960       }
4961     }
4962   }
4963 },
4964 },
4965 "parameters": {
4966   "interface-actuator" : {
4967     "in" : "query",
4968     "name" : "if",
4969     "type" : "string",
4970     "enum" : ["oic.if.a", "oic.if.baseline"]
4971   }
4972 },
4973 "definitions": {
4974   "RetrieveSchema" : {
4975     "properties": {
4976       "mode": {
4977         "description": "Current mode of device.",
4978         "type": "integer",

```

```

4981         "x-ocf-conversion": {
4982             "x-from-ocf": [
4983                 "modearray = [Auto,Circulation,Continuous]",
4984                 "mode = indexof modeArray[ocf.mode[0]]"
4985             ],
4986             "x-ocf-alias": "oic.r.mode",
4987             "x-to-ocf": [
4988                 "modearray = [Auto,Circulation,Continuous]",
4989                 "ocf.mode[0] = modearray[mode]"
4990             ]
4991         },
4992     },
4993     "supportedmodes": {
4994         "description": "Array of supported modes",
4995         "items": {
4996             "type": "integer"
4997         },
4998         "type": "array",
4999         "x-ocf-conversion": {
5000             "x-from-ocf": [
5001                 "modearray = [Auto,Circulation,Continuous]",
5002                 "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
5003 modearray[ocf.supportedmodes[x]]"
5004             ],
5005             "x-ocf-alias": "oic.r.mode",
5006             "x-to-ocf": [
5007                 "modearray = [Auto,Circulation,Continuous]",
5008                 "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
5009 modearray[supportedmodes[x]]"
5010             ]
5011         },
5012     },
5013 },
5014 "required": [
5015     "mode",
5016     "supportedmodes"
5017 ],
5018 "type": "object"
5019 }
5020
5021 ,
5022 "UpdateSchema" :
5023 {
5024     "properties": {
5025         "mode": {
5026             "description": "Current mode of device.",
5027             "type": "integer",
5028             "x-ocf-conversion": {
5029                 "x-from-ocf": [
5030                     "modearray = [Auto,Circulation,Continuous]",
5031                     "mode = indexof modeArray[ocf.mode[0]]"
5032                 ],
5033                 "x-ocf-alias": "oic.r.mode",
5034                 "x-to-ocf": [
5035                     "modearray = [Auto,Circulation,Continuous]",
5036                     "ocf.mode[0] = modearray[mode]"
5037                 ]
5038             },
5039         },
5040         "supportedmodes": {
5041             "description": "Array of supported modes",
5042             "items": {
5043                 "type": "integer"
5044             },
5045             "type": "array",
5046             "x-ocf-conversion": {
5047                 "x-from-ocf": [
5048                     "modearray = [Auto,Circulation,Continuous]",
5049                     "for x=0, x < sizeof(supportedmodes): supportedmodes[x] = indexof
5050 modearray[ocf.supportedmodes[x]]"
5051                 ],

```

```

5052         "x-ocf-alias": "oic.r.mode",
5053         "x-to-ocf": [
5054             "modearray = [Auto,Circulation,Continuous]",
5055             "for x=0, x < sizeof(supportedmodes): ocf.supportedmodes[x] =
5056 modearray[supportedmodes[x]]"
5057         ]
5058     },
5059     },
5060 },
5061 "required": [
5062     "mode",
5063     "supportedmodes"
5064 ],
5065 "type": "object"
5066 }
5067 }
5068 }
5069 }
5070

```

5071 A.11.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
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]]	지원되는 모드의 배열
mode	oic.r.mode	modearray = [Auto,Circulation,Continuous]ocf. mode[0] = modearray[mode]	modearray = [Auto,Circulation,Continuous]mode = indexof modeArray[ocf.mode[0]]	device 의 현재 모드

5072 A.11.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/HvacFanModeResURI		get	post		

5073 A.12 On Off 매핑

5074 A.12.1 개요

5075 이 API 는 OCF Binary Switch Resource 의 인스턴스와 AllJoyn Interface 집합 간의 매핑을 정의한다.
5076 검색된 Binary Switch 의 인스턴스는 항상 Operation.OnOffStatus interface 에 매핑된다.

5077
5078 Binary Switch 의 RETRIEVE 가 Operation.OnOffStatus Interface 의 인스턴스의 동작에 매핑된다.

5079
5080 Binary Switch 의 UPDATE 가 Operation.OnControl 또는 OffControl 의 방법 적용에 매핑된다.

5081
5082 value = true 가 Operation.OnControl 에 매핑된다.

5083 value = false 가 Operation.OffControl 에 매핑된다.

5084

5085 A.12.2 URI 예

5086 /OnOffResURI

5087 A.12.3 Resource Type

5088 resource type (rt)는 ['oic.r.switch.binary']로 정의된다.

5089 A.12.4 Swagger2.0 정의

```
5090 {
5091   "swagger": "2.0",
5092   "info": {
5093     "title": "On Off Mapping",
5094     "version": "OCFv1.0.0-20170317",
5095     "license": {
5096       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
5097       "x-description": "Redistribution and use in source and binary forms, with or without
5098 modification, are permitted provided that the following conditions are met:\n      1.
5099 Redistributions of source code must retain the above copyright notice, this list of conditions and
5100 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
5101 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
5102 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
5103 Connectivity Foundation, INC. \AS IS\ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
5104 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
5105 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
5106 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
5107 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
5108 SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND ON
5109 ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
5110 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
5111 OF SUCH DAMAGE.\n"
5112     },
5113   },
5114   "schemes": ["http"],
5115   "consumes": ["application/json"],
5116   "produces": ["application/json"],
5117   "paths": {
5118     "/OnOffResURI" : {
5119       "get": {
5120         "description": "This API defines the mapping between an instance of an OCF Binary Switch
5121 Resource and the\nequivalent Interface set by AllJoyn. A discovered instance of a Binary Switch is
5122 always\nmapped to an Operation.OnOffStatus interface.\nA RETRIEVE on a Binary Switch maps to an
5123 action on an instance of an Operation.OnOffStatus Interface.\nAn UPDATE on a Binary Switch maps to
5124 a method invocation on either Operation.OnControl or OffControl.\n value = true maps to
5125 Operation.OnControl\n value = false maps to Operation.OffControl\n",
5126         "parameters": [
5127           { "$ref": "#/parameters/interface-all" }
5128         ],
5129         "responses": {
5130           "200": {
5131             "description": "",
5132             "x-example": {
5133               "rt": ["oic.r.switch.binary"]
5134             },
5135             "schema": { "$ref": "#/definitions/RetrieveSchema" }
5136           }
5137         }
5138       },
5139     },
5140   },
5141   "post": {
5142     "description": "",
5143     "parameters": [
5144       { "$ref": "#/parameters/interface-actuator" },
5145       {
5146         "name": "body",
5147         "in": "body",
5148       }
5149     ]
5150   }
5151 }
```

```

5148         "required": true,
5149         "schema": { "$ref": "#/definitions/UpdateSchema" }
5150     }
5151 ],
5152 "responses": {
5153     "200": {
5154         "description": "",
5155         "schema": { "$ref": "#/definitions/UpdateSchema" }
5156     }
5157 }
5158 }
5159 },
5160 },
5161 "parameters": {
5162     "interface-actuator" : {
5163         "in" : "query",
5164         "name" : "if",
5165         "type" : "string",
5166         "enum" : ["oic.if.a", "oic.if.baseline"]
5167     },
5168     "interface-all" : {
5169         "in" : "query",
5170         "name" : "if",
5171         "type" : "string",
5172         "enum" : ["oic.if.s", "oic.if.a", "oic.if.baseline"]
5173     }
5174 },
5175 "definitions": {
5176     "RetrieveSchema" :
5177     {
5178         "properties": {
5179             "onoff": {
5180                 "description": "On/Off status of the device",
5181                 "type": "boolean",
5182                 "x-ocf-conversion": {
5183                     "x-from-ocf": [
5184                         "onoff = value"
5185                     ],
5186                     "x-ocf-alias": "oic.r.switch.binary",
5187                     "x-to-ocf": [
5188                         "value = onoff"
5189                     ]
5190                 }
5191             }
5192         },
5193         "required": [
5194             "onoff"
5195         ],
5196         "type": "object"
5197     }
5198 },
5199 "UpdateSchema" :
5200 {
5201     "properties": {
5202         "switchon": {
5203             "description": "Turn on the device",
5204             "format": "method",
5205             "type": "string",
5206             "x-ocf-conversion": {
5207                 "x-ocf-alias": "oic.r.switch.binary",
5208                 "x-to-ocf": [
5209                     "value = true"
5210                 ]
5211             }
5212         }
5213     },
5214     "type": "object"
5215 }
5216 }
5217 }
5218 }

```

5219 }
5220

5221 A.12.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
onoff	oic.r.switch.binary	value = onoff	onoff = value	device 의 On/Off 상태

5222 A.12.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/OnOffResURI		get	post		

5223 A.13 Oven Cycle Phase 매핑

5224 A.13.1 개요

5225 이 API 는 AllJoyn OvenCyclePhase interface 의 인스턴스와 OCF OperationalState Resource 간의
5226 매핑을 정의한다.

5227 OvenCyclePhase Property 는 값 0x00-0x7F, 0x80-0xFF 를 사전 정의한다. 여기에 정의된 매핑은
5228 스펙 정의 값에만 영향을 미친다.

5229 모든 제조사 정의 값은 OCF 에서 제조사 정의 Property 를 위한 x.<organization> 구문을 사용해서
5230 표현된다.

5231 AllJoyn interface 는 Method, GetVendorPhasesDescription 도 지원하며, 이는 OCF 에서
5232 oic.r.action collection 내의 oic.r.actuator 의 인스턴스를 사용해서 처리된다.

5233 자세한 사항은 매핑 스펙의 섹션 8 을 참조하기 바란다.

5234

5235 A.13.2 URI 예

5236 /OvenCyclePhaseResURI

5237 A.13.3 Resource Type

5238 resource type (rt)는 ['oic.r.operationalstate']로 정의된다.

5239 A.13.4 Swagger2.0 정의

```
5240 {  
5241   "swagger": "2.0",  
5242   "info": {  
5243     "title": "Oven Cycle Phase Mapping",  
5244     "version": "OCFv1.0.0-20170317",  
5245     "license": {  
5246       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",  
5247       "x-description": "Redistribution and use in source and binary forms, with or without  
5248 modification, are permitted provided that the following conditions are met:\n          1.  
5249 Redistributions of source code must retain the above copyright notice, this list of conditions and  
5250 the following disclaimer.\n          2. Redistributions in binary form must reproduce the above  
5251 copyright notice, this list of conditions and the following disclaimer in the documentation and/or  
5252 other materials provided with the distribution.\n          THIS SOFTWARE IS PROVIDED BY THE Open  
5253 Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT  
5254 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR  
5255 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n          IN NO EVENT SHALL THE Open Connectivity  
5256 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
```

```

5257 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
5258 SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n          HOWEVER CAUSED AND ON
5259 ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
5260 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
5261 OF SUCH DAMAGE.\n"
5262     }
5263   },
5264   "schemes": ["http"],
5265   "consumes": ["application/json"],
5266   "produces": ["application/json"],
5267   "paths": {
5268     "/OvenCyclePhaseResURI" : {
5269       "get": {
5270         "description": "This API defines the mapping between an instance of an AllJoyn
5271 OvenCyclePhase interface and the OCF OperationalState Resource.\nOvenCyclePhase cyclephase Property
5272 pre-defines values 0x00-0x7F, 0x80-0xFF is for vendor specific values. The mapping defined herein
5273 covers only Spec defined values.\nAny vendor defined value shall be represented in OCF using the
5274 x.<organization> syntax for a vendor defined Property.\nThe AllJoyn interface also supports a
5275 Method, GetVendorPhasesDescription; this is handled in OCF using an instance of oic.r.actuator
5276 within an oic.r.action collection.\nPlease see Section 8 of the Mapping Specification for
5277 specifics.\n",
5278         "parameters": [
5279           { "$ref": "#/parameters/interface-sensor" }
5280         ],
5281         "responses": {
5282           "200": {
5283             "description": "",
5284             "x-example": {
5285               "rt": ["oic.r.operationalstate"]
5286             },
5287             "schema": { "$ref": "#/definitions/RetrieveSchema" }
5288           }
5289         }
5290       }
5291     }
5292   },
5293   },
5294   },
5295   "parameters": {
5296     "interface-sensor" : {
5297       "in" : "query",
5298       "name" : "if",
5299       "type" : "string",
5300       "enum" : ["oic.if.s", "oic.if.baseline"]
5301     }
5302   },
5303   "definitions": {
5304     "RetrieveSchema" :
5305     {
5306       "properties": {
5307         "cyclephase": {
5308           "description": "Current phase of the operational cycle",
5309           "type": "integer",
5310           "x-ocf-conversion": {
5311             "x-from-ocf": [
5312               "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
5313               "cyclephase = indexof statearray[currentmachinestate[0]]"
5314             ],
5315             "x-ocf-alias": "oic.r.operationalstate",
5316             "x-to-ocf": [
5317               "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
5318               "currentmachinestate = phasearray[cyclephase]"
5319             ]
5320           }
5321         },
5322         "getvendorphasesdescription": {
5323           "description": "Get cycle phases description",
5324           "x-ocf-conversion": {
5325             "x-ocf-alias": "oic.r.action"
5326           },
5327           "x-ocf-type": "method"

```



```

5328     },
5329     "supportedcyclephases": {
5330         "description": "Array of cycle phases supported by the Appliance.",
5331         "items": {
5332             "type": "integer"
5333         },
5334         "type": "array",
5335         "x-ocf-conversion": {
5336             "x-from-ocf": [
5337                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
5338                 "for x=0, x < sizeof(machinestates): supportedcyclephases[x] = indexof
5339 phasearray[machinestates[x]]"
5340             ],
5341             "x-ocf-alias": "oic.r.operationalstate",
5342             "x-to-ocf": [
5343                 "phasearray = [Unavailable,Preheating,Cooking,Cleaning]",
5344                 "for x=0, x < sizeof(supportedcyclephases): machinestates[x] =
5345 phasearray[supportedcyclephases[x]]"
5346             ]
5347         },
5348     },
5349 },
5350 "required": [
5351     "cyclephase",
5352     "supportedcyclephases"
5353 ],
5354 "type": "object"
5355 }
5356 }
5357 }
5358 }
5359 }

```

5360 A.13.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
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]]	기기에 의해 지원되는 주기 위상의 배열
cyclephase	oic.r.operationalstate	phasearray = [Unavailable,Preheating,Cooking,Cleaning]currentmachinestate = phasearray[cyclephase]	phasearray = [Unavailable,Preheating,Cooking,Cleaning]cyclephase = indexof statearray[currentmachinestate[0]]	동작 주기의 현재 위상
getvendorphasedescription	oic.r.action			주기 위상 설명 취득

5361 A.13.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/OvenCyclePhaseResURI		get			

A.14 Target Humidity 매핑

A.14.1 개요

이 API 는 AllJoyn TargetHumidity Interface 의 인스턴스와 OCF Resource 간의 매핑을 정의한다. Humidity Sensor 의 POST 가 Environment.TargetHumidity Interface 의 인스턴스의 동작에 매핑된다.

A.14.2 URI 예

/TargetHumidityResURI

A.14.3 Resource Type

resource type (rt)는 ['oic.r.humidity', 'oic.r.selectablelevels']로 정의된다.

A.14.4 Swagger2.0 정의

```
{
  "swagger": "2.0",
  "info": {
    "title": "Target 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": {
    "/TargetHumidityResURI" : {
      "get": {
        "description": "This API defines the mapping between an instance of an AllJoyn
TargetHumidity Interface and the OCF Resource Equivalent.\nA POST on a Humidity Sensor maps to an
action on an instance of an Environment.TargetHumidity Interface.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-actuator"}
        ],
        "responses": {
          "200": {
            "description": "",
            "x-example": {
              "rt": ["oic.r.humidity", "oic.r.selectablelevels"]
            },
            "schema": { "$ref": "#/definitions/RetrieveSchema" }
          }
        }
      }
    }
  }
}
```

```

5420     }
5421   },
5422   "post": {
5423     "description": "",
5424     "parameters": [
5425       { "$ref": "#/parameters/interface-actuator" },
5426       {
5427         "name": "body",
5428         "in": "body",
5429         "required": true,
5430         "schema": { "$ref": "#/definitions/UpdateSchema" }
5431       }
5432     ],
5433     "responses": {
5434       "200": {
5435         "description": "",
5436         "x-example": {
5437           "rt": ["oic.r.humidity", "oic.r.selectablelevels"]
5438         },
5439         "schema": { "$ref": "#/definitions/UpdateSchema" }
5440       }
5441     }
5442   }
5443 }
5444 }
5445 },
5446 "parameters": {
5447   "interface-actuator" : {
5448     "in" : "query",
5449     "name" : "if",
5450     "type" : "string",
5451     "enum" : ["oic.if.a", "oic.if.baseline"]
5452   }
5453 },
5454 "definitions": {
5455   "RetrieveSchema" :
5456   {
5457     "properties": {
5458       "maxvalue": {
5459         "type": "number",
5460         "x-ocf-conversion": {
5461           "x-from-ocf": [
5462             "maxvalue = range[1]",
5463             "otherwise: maxvalue = 100"
5464           ],
5465           "x-ocf-alias": "oic.r.humidity",
5466           "x-to-ocf": [
5467             "range[1] = maxvalue"
5468           ]
5469         }
5470       },
5471       "minvalue": {
5472         "type": "number",
5473         "x-ocf-conversion": {
5474           "x-from-ocf": [
5475             "minvalue = range[0]",
5476             "otherwise: minvalue = 0"
5477           ],
5478           "x-ocf-alias": "oic.r.humidity",
5479           "x-to-ocf": [
5480             "range[0] = minvalue"
5481           ]
5482         }
5483       }
5484     },
5485     "selectablehumiditylevels": {
5486       "items": {
5487         "type": "number"
5488       },
5489       "type": "array",
5490       "x-ocf-conversion": {

```

```

5491         "x-from-ocf": [
5492             "selectablehumiditylevels[] = availablelevels[]"
5493         ],
5494         "x-ocf-alias": "oic.r.selectablelevels",
5495         "x-to-ocf": [
5496             "availablelevels[] = selectablehumiditylevels[]"
5497         ]
5498     },
5499 },
5500 "stepvalue": {
5501     "type": "number",
5502     "x-ocf-conversion": {
5503         "x-from-ocf": [
5504             "stepvalue = step",
5505             "otherwise: stepvalue = 1"
5506         ],
5507         "x-ocf-alias": "oic.r.humidity",
5508         "x-to-ocf": [
5509             "step = stepvalue"
5510         ]
5511     }
5512 },
5513 "targetvalue": {
5514     "description": "Measured value",
5515     "type": "number",
5516     "x-ocf-conversion": {
5517         "x-from-ocf": [
5518             "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
5519             "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
5520         ],
5521         "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
5522         "x-to-ocf": [
5523             "if minvalue != maxvalue, ocf.desiredhumidity = targetvalue;ocf.targetlevel =
5524 selectablehumiditylevels[0].",
5525             "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
5526         ]
5527     }
5528 },
5529 },
5530 "required": [
5531     "targetvalue",
5532     "minvalue",
5533     "maxvalue",
5534     "stepvalue",
5535     "selectablehumiditylevels"
5536 ],
5537 "type": "object"
5538 }
5539
5540 ,
5541 "UpdateSchema" :
5542 {
5543     "properties": {
5544         "maxvalue": {
5545             "type": "number",
5546             "x-ocf-conversion": {
5547                 "x-from-ocf": [
5548                     "maxvalue = range[1]",
5549                     "otherwise: maxvalue = 100"
5550                 ],
5551                 "x-ocf-alias": "oic.r.humidity",
5552                 "x-to-ocf": [
5553                     "range[1] = maxvalue"
5554                 ]
5555             }
5556         },
5557         "minvalue": {
5558             "type": "number",
5559             "x-ocf-conversion": {
5560                 "x-from-ocf": [
5561                     "minvalue = range[0]",

```

```

5562         "otherwise: minvalue = 0"
5563     ],
5564     "x-ocf-alias": "oic.r.humidity",
5565     "x-to-ocf": [
5566         "range[0] = minvalue"
5567     ]
5568 },
5569 },
5570 "selectablehumiditylevels": {
5571     "items": {
5572         "type": "number"
5573     },
5574     "type": "array",
5575     "x-ocf-conversion": {
5576         "x-from-ocf": [
5577             "selectablehumiditylevels[] = availablelevels[]"
5578         ],
5579         "x-ocf-alias": "oic.r.selectablelevels",
5580         "x-to-ocf": [
5581             "availablelevels[] = selectablehumiditylevels[]"
5582         ]
5583     }
5584 },
5585 "stepvalue": {
5586     "type": "number",
5587     "x-ocf-conversion": {
5588         "x-from-ocf": [
5589             "stepvalue = step",
5590             "otherwise: stepvalue = 1"
5591         ],
5592         "x-ocf-alias": "oic.r.humidity",
5593         "x-to-ocf": [
5594             "step = stepvalue"
5595         ]
5596     }
5597 },
5598 "targetvalue": {
5599     "description": "Measured value",
5600     "type": "number",
5601     "x-ocf-conversion": {
5602         "x-from-ocf": [
5603             "if x-ocf-alias == oic.r.humidity, targetvalue = desiredhumidity.",
5604             "if x-ocf-alias == oic.r.selectablelevels, targetvalue = targetlevel."
5605         ],
5606         "x-ocf-alias": "oic.r.humidity,oic.r.selectablelevels",
5607         "x-to-ocf": [
5608             "if minvalue != maxvalue, ocf.desiredhumidity = targetvalue;ocf.targetlevel =
5609 selectablehumiditylevels[0].",
5610             "if minvalue == maxvalue, ocf.targetlevel = targetvalue."
5611         ]
5612     }
5613 },
5614 },
5615 "required": [
5616     "targetvalue",
5617     "minvalue",
5618     "maxvalue",
5619     "stepvalue",
5620     "selectablehumiditylevels"
5621 ],
5622 "type": "object"
5623 }
5624 }
5625 }
5626 }
5627

```

A.14.5 Property 정의

['AllJoyn']	OCF Resource	To OCF	From OCF	설명
-------------	--------------	--------	----------	----

Property name				
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.	측정치
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	

5629 A.14.6 CRUDN 동작

Resource	Create	Read	Update	Delete	Notify
/TargetHumidityResURI		get	post		

5630 A.15 Target Temperature 매핑

5631 A.15.1 개요

5632 이 API 는 sensor interface 에 있는 OCF Temperature 인스턴스와 AllJoyn Current Temperature
5633 interface 간의 매핑을 정의한다.

5634 온도 센서의 RETRIEVE 가 Environment.CurrentTemperature Interface 인스턴스의 동작에
5635 매핑된다.

5636
5637

5638 A.15.2 URI 예

5639 /TargetTemperatureResURI

5640 A.15.3 Resource Type

5641 resource type (rt)는 ['oic.r.temperature']로 정의된다.

5642 A.15.4 Swagger2.0 정의

```
5643 {
5644   "swagger": "2.0",
5645   "info": {
5646     "title": "Target Temperature Mapping",
5647     "version": "OCFv1.0.0-20170317",
5648     "license": {
```

```

5649         "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
5650         "x-description": "Redistribution and use in source and binary forms, with or without
5651 modification, are permitted provided that the following conditions are met:\n          1.
5652 Redistributions of source code must retain the above copyright notice, this list of conditions and
5653 the following disclaimer.\n          2. Redistributions in binary form must reproduce the above
5654 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
5655 other materials provided with the distribution.\n\n          THIS SOFTWARE IS PROVIDED BY THE Open
5656 Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
5657 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
5658 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n          IN NO EVENT SHALL THE Open Connectivity
5659 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
5660 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
5661 SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n          HOWEVER CAUSED AND ON
5662 ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
5663 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
5664 OF SUCH DAMAGE.\n"
5665     }
5666 },
5667 "schemes": ["http"],
5668 "consumes": ["application/json"],
5669 "produces": ["application/json"],
5670 "paths": {
5671     "/TargetTemperatureResURI" : {
5672         "get": {
5673             "description": "This API defines the mapping between an instance of an OCF Temperature
5674 which exposes only a sensor interface\nand the AllJoyn Current Temperature interface.\nA RETRIEVE
5675 on a Temperature Sensor maps to an action on an instance of an Environment.CurrentTemperature
5676 Interface.\n",
5677             "parameters": [
5678                 { "$ref": "#/parameters/interface-actuator" }
5679             ],
5680             "responses": {
5681                 "200": {
5682                     "description": "",
5683                     "x-example":
5684                         {
5685                             "rt": ["oic.r.temperature"]
5686                         },
5687                     "schema": { "$ref": "#/definitions/RetrieveSchema" }
5688                 }
5689             }
5690         },
5691     },
5692     "post": {
5693         "description": "",
5694         "parameters": [
5695             { "$ref": "#/parameters/interface-actuator" },
5696             {
5697                 "name": "body",
5698                 "in": "body",
5699                 "required": true,
5700                 "schema": { "$ref": "#/definitions/UpdateSchema" }
5701             }
5702         ],
5703         "responses": {
5704             "200": {
5705                 "description": "",
5706                 "schema": { "$ref": "#/definitions/UpdateSchema" }
5707             }
5708         }
5709     }
5710 },
5711 },
5712 "parameters": {
5713     "interface-actuator" : {
5714         "in" : "query",
5715         "name" : "if",
5716         "type" : "string",
5717         "enum" : ["oic.if.a", "oic.if.baseline"]
5718     }
5719 },

```

```

5720 "definitions": {
5721     "RetrieveSchema" :
5722     {
5723         "properties": {
5724             "maxvalue": {
5725                 "type": "number",
5726                 "x-ocf-conversion": {
5727                     "x-from-ocf": [
5728                         "maxvalue = range[1]",
5729                         "otherwise: maxvalue = MAXINT"
5730                     ],
5731                     "x-ocf-alias": "oic.r.temperature",
5732                     "x-to-ocf": [
5733                         "range[1] = maxvalue"
5734                     ]
5735                 }
5736             },
5737             "minvalue": {
5738                 "type": "number",
5739                 "x-ocf-conversion": {
5740                     "x-from-ocf": [
5741                         "minvalue = range[0]",
5742                         "otherwise: minvalue = -MAXINT"
5743                     ],
5744                     "x-ocf-alias": "oic.r.temperature",
5745                     "x-to-ocf": [
5746                         "range[0] = minvalue"
5747                     ]
5748                 }
5749             },
5750             "step": {
5751                 "type": "number",
5752                 "x-ocf-conversion": {
5753                     "x-from-ocf": [
5754                         "step = ocf.step",
5755                         "otherwise: step = undefined (0x00)"
5756                     ],
5757                     "x-ocf-alias": "oic.r.temperature",
5758                     "x-to-ocf": [
5759                         "ocf.step = step"
5760                     ]
5761                 }
5762             },
5763             "targetvalue": {
5764                 "description": "Measured value",
5765                 "type": "number",
5766                 "x-ocf-conversion": {
5767                     "x-from-ocf": {
5768                         "oneOf": [
5769                             {
5770                                 "properties": {
5771                                     "enum": [
5772                                         "C"
5773                                     ],
5774                                     "units": "string"
5775                                 },
5776                                 "x-from-ocf": [
5777                                     "targetvalue = temperature"
5778                                 ]
5779                             },
5780                             {
5781                                 "properties": {
5782                                     "enum": [
5783                                         "F"
5784                                     ],
5785                                     "units": "string"
5786                                 },
5787                                 "x-from-ocf": [
5788                                     "targetvalue = (temperature-32)*5/9"
5789                                 ]
5790                             }
5791                         ]
5792                     }
5793                 }
5794             }
5795         }
5796     }
5797 }

```



```

5791         {
5792             "properties": {
5793                 "enum": [
5794                     "K"
5795                 ],
5796                 "units": "string"
5797             },
5798             "x-from-ocf": [
5799                 "targetvalue = temperature-273.15"
5800             ]
5801         }
5802     ],
5803 },
5804 "x-ocf-alias": "oic.r.temperature",
5805 "x-to-ocf": [
5806     "temperature = targetvalue",
5807     "units = C"
5808 ]
5809 }
5810 },
5811 },
5812 "required": [
5813     "targetvalue",
5814     "minvalue",
5815     "maxvalue",
5816     "step"
5817 ],
5818 "type": "object"
5819 }
5820
5821 ,
5822 "UpdateSchema" :
5823 {
5824     "properties": {
5825         "maxvalue": {
5826             "type": "number",
5827             "x-ocf-conversion": {
5828                 "x-from-ocf": [
5829                     "maxvalue = range[1]",
5830                     "otherwise: maxvalue = MAXINT"
5831                 ],
5832                 "x-ocf-alias": "oic.r.temperature",
5833                 "x-to-ocf": [
5834                     "range[1] = maxvalue"
5835                 ]
5836             }
5837         },
5838         "minvalue": {
5839             "type": "number",
5840             "x-ocf-conversion": {
5841                 "x-from-ocf": [
5842                     "minvalue = range[0]",
5843                     "otherwise: minvalue = -MAXINT"
5844                 ],
5845                 "x-ocf-alias": "oic.r.temperature",
5846                 "x-to-ocf": [
5847                     "range[0] = minvalue"
5848                 ]
5849             }
5850         },
5851         "step": {
5852             "type": "number",
5853             "x-ocf-conversion": {
5854                 "x-from-ocf": [
5855                     "step = ocf.step",
5856                     "otherwise: step = undefined (0x00)"
5857                 ],
5858                 "x-ocf-alias": "oic.r.temperature",
5859                 "x-to-ocf": [
5860                     "ocf.step = step"
5861                 ]

```

```

5862     }
5863   },
5864   "targetvalue": {
5865     "description": "Measured value",
5866     "type": "number",
5867     "x-ocf-conversion": {
5868       "x-from-ocf": {
5869         "oneOf": [
5870           {
5871             "properties": {
5872               "enum": [
5873                 "C"
5874               ],
5875               "units": "string"
5876             },
5877             "x-from-ocf": [
5878               "targetvalue = temperature"
5879             ]
5880           },
5881           {
5882             "properties": {
5883               "enum": [
5884                 "F"
5885               ],
5886               "units": "string"
5887             },
5888             "x-from-ocf": [
5889               "targetvalue = (temperature-32)*5/9"
5890             ]
5891           },
5892           {
5893             "properties": {
5894               "enum": [
5895                 "K"
5896               ],
5897               "units": "string"
5898             },
5899             "x-from-ocf": [
5900               "targetvalue = temperature-273.15"
5901             ]
5902           }
5903         ]
5904       },
5905       "x-ocf-alias": "oic.r.temperature",
5906       "x-to-ocf": [
5907         "temperature = targetvalue",
5908         "units = C"
5909       ]
5910     }
5911   },
5912   "required": [
5913     "targetvalue",
5914     "minvalue",
5915     "maxvalue",
5916     "step"
5917   ],
5918   "type": "object"
5919 }
5920 }
5921 }
5922 }
5923 }
5924

```

5925 A.15.5 Property 정의

['AllJoyn'] Property name	OCF Resource	To OCF	From OCF	설명
targetvalue	oic.r.temperature	temperature = targetvalueunits	oneOf	측정치

		= C		
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	

5926 **A.15.6 CRUDN 동작**

Resource	Create	Read	Update	Delete	Notify
/TargetTemperatureResURI		get	post		

5927
5928