OCF Wi-Fi Easy Setup Specification

VERSION 2.1.1 | February 2020

OPEN CONNECTIVITY
FOUNDATION™

2   Legal Disclaimer

3

4   NOTHING CONTAINED IN THIS DOCUMENT SHALL BE DEEMED AS GRANTING YOU ANY KIND
5   OF LICENSE IN ITS CONTENT, EITHER EXPRESSLY OR IMPLIEDLY, OR TO ANY
6   INTELLECTUAL PROPERTY OWNED OR CONTROLLED BY ANY OF THE AUTHORS OR
7   DEVELOPERS OF THIS DOCUMENT. THE INFORMATION CONTAINED HEREIN IS PROVIDED
8   ON AN "AS IS" BASIS, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW,
9   THE AUTHORS AND DEVELOPERS OF THIS SPECIFICATION HEREBY DISCLAIM ALL OTHER
10   WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT
11   COMMON LAW, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF
12   MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. OPEN CONNECTIVITY
13   FOUNDATION, INC. FURTHER DISCLAIMS ANY AND ALL WARRANTIES OF NON-
14   INFRINGEMENT, ACCURACY OR LACK OF VIRUSES.

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

17   Copyright © 2017-2020 Open Connectivity Foundation, Inc. All rights reserved.

18   Copying or other form of reproduction and/or distribution of these works are strictly prohibited.
19

Copyright Open Connectivity Foundation, Inc. © 2017-2020. All rights Reserved

CONTENTS

80

81

Figures

Tables

1 Scope

This document defines functional extensions to the capabilities defined in ISO/IEC 30118-1:2018 to meet the requirements of Wi-Fi Easy Setup. It specifies new Resource Types to enable the functionality and any extensions to the existing capabilities defined in ISO/IEC 30118-1:2018.

2 Normative references

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

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

ISO/IEC 30118-2:2018 Information technology -- Open Connectivity Foundation (OCF) Specification -- Part 2: Security specification
https://www.iso.org/standard/74239.html
Latest version available at: https://openconnectivity.org/specs/OCF_Security_Specification.pdf

ISO/IEC 30118-5:2018 Information technology -- Open Connectivity Foundation (OCF) Specification -- Part 5: Smart home device specification
https://www.iso.org/standard/74242.html
Latest version available at: https://openconnectivity.org/specs/OCF_Device_Specification.pdf

IEEE 802.11:2016, IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, December 2016
https://standards.ieee.org/findstds/standard/802.11-2016.html

IETF RFC 5646, *Tags for Identifying Languages*, September 2009
https://www.rfc-editor.org/info/rfc5646

OpenAPI specification, aka *Swagger RESTful API Documentation Specification*, Version 2.0
https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md

3 Terms, definitions, and abbreviated terms

3.1 Terms and definitions

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

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

– ISO Online browsing platform: available at https://www.iso.org/obp

– IEC Electropedia: available at http://www.electropedia.org/

3.1.1
Easy Setup
process of configuring an Enrollee (3.1.3) using a Mediator (3.1.5) by transferring of essential information to the Enrollee (3.1.3)

3.1.2
Easy Setup Enrollment
step during Easy Setup in which the Enrollee (3.1.3) is contacted by the Mediator (3.1.5) to configure the Enroller's (3.1.4) information by means of accessing Easy Setup (3.1.1) Resources

3.1.3
Enrollee
device that needs to be configured and connected. E.g. Air-conditioner, Printer

3.1.4
Enroller
target network entity to which the Enrollee (3.1.3) connects. E.g. Wi-Fi AP

3.1.5
Mediator
logical function that enables the Enrollee (3.1.3) to connect to the target network (i.e. Enroller (3.1.4))

Note 1 to Entry: The Mediator transfers configuration information to the Enrollee. E.g. Mobile Phone

3.2 Abbreviated terms

3.2.1
CID
Company Identifier (ID)

3.2.2
IE
Information Element

3.2.3
Soft AP
Software Enabled Access Point

3.2.4
TLV
type-length-value

4 Document conventions and organization

4.1 Conventions

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

4.2 Notation

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

Required (or shall or mandatory)(M).

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

Recommended (or should)(S).

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

Allowed (may or allowed)(O).

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

DEPRECATED.

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

Conditionally allowed (CA)

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

Conditionally required (CR)

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

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

Words that are emphasized are printed in italic.

5   Overview

5.1   Introduction

This document describes a way to setup and configure a new OCF Device, using an already configured OCF Device or onboarding tool.

The described setup and configure mechanism is optional and other mechanisms are allowed to be used.

Specifically, this method allows the transferring of essential information to the new Device, which includes:

– Local network connection information, e.g. in case of Wi-Fi it will be Wi-Fi access point information.

– Device Configuration: Additional Device configuration information.

Easy Setup can be enhanced in future by incorporating other suitable technologies.

Annex A specifies the Resource Type definitions using the schema defined in the OpenAPI specification as the API definition language that shall be followed by an OCF Device realizing the Resources specified in this document.

5.2   Architecture

Figure 1 shows the deployment architectural approach.
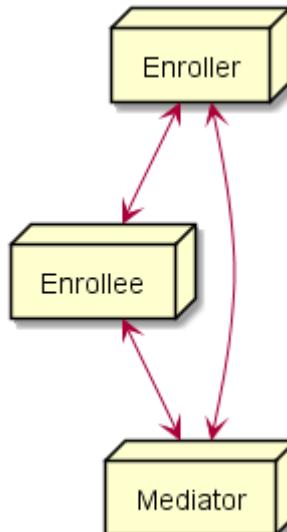
Figure 1 – Easy Setup deployment architecture

Easy Setup defines the following roles: Enrollee, Enroller, and Mediator. Please refer to clause 3 for the definitions thereof.

5.3   Example Scenario

The following scenario presents a typical setup case.

The configuration information and steps taken may vary depending on the Device's type and status.

249 1) The Enrollee enters Easy Setup mode (when the Device is unboxed for the first time, it may be
250   in this mode by default).

251 2) The Mediator discovers and connects to the Enrollee.

252 3) The Mediator performs Security Provisioning of the Enrollee.

253 4) The Mediator transmits Wi-Fi Setting Information to the Enrollee.

254 5) Using the information received from the Mediator, the Enrollee connects to the Enroller (Wi-Fi
255   AP).

256 6   Resource model

257 6.1   Introduction

258 Devices capable of Easy Setup shall support the following Resource Types.
259 1) EasySetup Resource Type

260 2) WiFiConf Resource Type

261 3) DevConf Resource Type

262 Instances of these Resources Type (Resources) shall be excluded in the IDD for the Introspection
263 Resource (see clause 11.4 in ISO/IEC 30118-1:2018).

264 The EasySetup Resource Type is a Collection Resource and shall contain Links to instances of at
265 least WiFiConf and DevConf. A vendor may add links to other Resource Types. The relationship
266 between the EasySetup Resource Type and linked Resources is shown in Figure 2.

267 NOTE   The EasySetup Resource Type supports the batch Interface (oic.if.b) which allows for efficient data delivery with
268 a single request rather than multiple requests to each linked Resource.
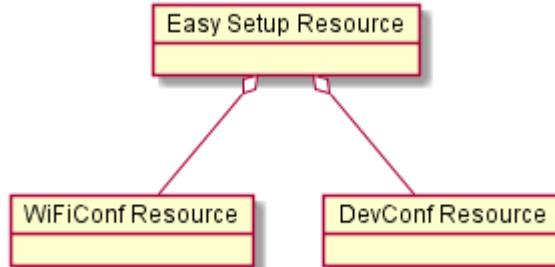
269

270 **Figure 2 – Easy Setup Resource Types**

271 6.2   EasySetup Resource

272 6.2.1 Overview

273 The EasySetup Resource stores useful information including current status of Enrollee and last
274 error code which was produced in the process of Easy Setup.

275 6.2.2 Resource

276 The Easy Setup Resource Type is as defined in Table 1.

277

Table 1 – EasySetup Resource Type

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
/example/Easy SetupResURI	EasySetup	oic.r.easysetup, oic.wk.col	oic.if.baseline, oic.if.ll, oic.if.b	Top level Resource for Easy Setup. Indicates easy setup status. The Resource properties exposed are listed in Table 2.	N/A

278

279 Table 2 defines the details for the "oic.r.easysetup" Resource Type.

280 **Table 2 – "oic.r.easysetup" Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Easy Setup Provisioning Status	ps	integer	enum	N/A	R	Yes	Easy setup provisioning status of the Device 0: Need to Setup, 1: Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller, 4~254: Reserved, 255: EOF
Last Error Code	lec	integer	enum	N/A	R	Yes	Indicates a failure reason if it fails to connect to Enroller 0: No error, 1: Given SSID is not found, 2: Wi-Fi password is wrong, 3: IP address is not allocated, 4: NO internet connection, 5: Timeout, 6: Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not supported by the Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller), 9: Wi-Fi Encryption Type is wrong (failure while connecting to the Enroller), 10~254: Reserved, 255: Unknown error.
Connect	cn	array of integer	N/A	N/A	RW	Yes	Array of connection types to trigger Enrollee to initiate connection: 1: Wi-Fi,

							2: Other transport to be added in a future (e.g. BLE))
Links	links	array	N/A	N/A	R	Yes	Array of links that are WiFiConf and DevConf Resource.

281

282 Enrollee shall set the following as default values (for example, when Device is unboxed first time):

283 – "ps" equal to 0.

284 – "lec" equal to 0.

285 – "cn" equal to an empty array.

6.3 WiFiConf Resource Type

6.3.1 Introduction

288 The WiFiConf Resource Type stores information to help an Enrollee to connect to an existing Wi-
289 Fi AP.

6.3.2 Resource Type

291 The WiFiConf Resource Type is as defined in Table 3.

292 **Table 3 – WiFiConf Resource Type**

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
/example/WiFiConfResURI	WiFiConf	oic.r.wificonf	oic.if.baseline+ oic.if.rw	Contains Wi-Fi related properties The Resource properties exposed are listed in Table 4.	N/A

293

294 Table 4 defines the details for the "oic.r.wificonf" Resource Type.

295 **Table 4 – "oic.r.wificonf" Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Supported Wi-Fi Mode Type	swmt	array of string	enum	N/A	R	Yes	Supported Wi-Fi modes by Enrollee. Can be multiple. ("A", "B", "G","N", "AC")
Supported Wi-Fi Frequency	swf	array of string	Refer to description for valid values.	N/A	R	Yes	Supported Wi-Fi frequencies by Enrollee. Can be multiple. ("2.4G", "5G")
Target Network Name	tnn	string	N/A	N/A	RW	Yes	Target network name (SSID of Wi-Fi AP i.e. enroller)
Credential	cd	string	N/A	N/A	RW	No	Credential information of Wi-Fi AP (Password used to connect to enroller).

Wi-Fi Auth Type	wat	string	enum	N/A	RW	Yes	Wi-Fi auth type ("None", "WEP", "WPA_PSK", "WPA2_PSK")
Wi-Fi Encryption Type	wet	string	enum	N/A	RW	Yes	Wi-Fi encryption type ("None", "WEP_64", "WEP_128", "TKIP", "AES", "TKIP_AES")
Supported Wi-Fi Auth Type	swat	array of string	enum	N/A	R	Yes	Supported Wi-Fi Auth types. Can be multiple. ("None", "WEP", "WPA_PSK", "WPA2_PSK")
Supported Wi-Fi Encryption Type	swet	array of string	enum	N/A	R	Yes	Supported Wi-Fi Encryption types. Can be multiple. ("None", "WEP-64", "WEP_128", "TKIP", "AES", "TKIP_AES")

296

6.4 DevConf Resource Type

6.4.1 Introduction

299 The DevConf Resource Type stores Device configuration information required in Wi-Fi Easy Setup.

6.4.2 Resource Type

301 The DevConf Resource Type is as defined in Table 5

302   **Table 5 – DevConf Resource Type**

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
/example/Dev ConfResURI	DevConf	oic.r.devconf	oic.if.baseline, "oic.if.r"	Stores device configuration information required in Easy Setup process The Resource properties exposed are listed in Table 6.	N/A

303

304 Table 6 defines the details for the "oic.r.devconf" Resource Type.

305   **Table 6 – "oic.r.devconf" Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Device Name	dn	one of: string or array of object	N/A	N/A	R	Yes	Indicates a pre-configured device name in language indicated by "dl" in "/oic/con". or An array of objects where each object has a language field (containing an IETF RFC 5646 language tag) and a value field

| | | | | | | | containing the pre-configured device name in the indicated language. |
| | | | | | | | The pre-configured device name is presented by enrollee to mediator during easy-setup process. |

306

7 Network and connectivity

Both the Mediator and Enrollee communicate via a common connectivity (e.g. Wi-Fi).

If using Wi-Fi for Easy Setup then the Enrollee shall have capability to act as a Soft AP. A Soft AP shall support the access point requirements defined by IEEE 802.11:2016.

8 Functional interactions

8.1 Onboarding, Provisioning and Configuration

The Mediator may be present as a standalone function or in conjunction with other functions or services such as AMS as part of an OBT (Onboarding Tool); please refer to the ISO/IEC 30118-2:2018.

8.2 Resource discovery

The Mediator connects to the Enrollee via a mutually supported connection.

When in Easy Setup phase, if using Wi-Fi as the connectivity between the Enrollee and the Mediator then the Enrollee shall make itself discoverable as a Soft AP. The Soft AP has additional availability constraints which are documented in ISO/IEC 30118-2:2018.

8.3 Retrieving and Updating Easy Setup Resources

The Enrollee shall expose Easy Setup Resources such that a Mediator is able to discover them using standard OCF Resource discovery methods (i.e. via a RETRIEVE on /oic/res); see the ISO/IEC 30118-1:2018, clause 11.3.

Easy Setup Resources shall expose only secure Endpoints (e.g. CoAPS); see the ISO/IEC 30118-1:2018, clause 10.

The Mediator may RETRIEVE a Resource within the Easy Setup Collection or the Collection itself to check the Enrollee's status at any stage of Easy Setup. This applies only when the Enrollee & the Mediator are on a common network.

The Mediator may UPDATE Resource Property(-ies) on the Enrollee. Upon receipt of the request from the Mediator the Enrollee shall update its current Resource Property Values, and shall perform any required action. For example, if the "cn" Property of "EasySetup" Resource is updated by the Mediator, to indicate connection to Wi-Fi, the Enrollee shall start the connection to Enroller.

For details of Easy Setup Resources refer to clause 6.

8.4 Error Handling

The "lec" Property of the EasySetup Resource (i.e. "oic.r.easysetup") is used to indicate the error that occurred in the Easy Setup process while trying to connect to the Enroller (using the information provided by the Mediator in WiFiConf Resource):

– The Enrollee shall set "lec" Property to 1, if it fails to connect because it can't find the SSID.

– The Enrollee shall set "lec" Property to 2, if it fails to connect due to wrong credential (password) information.

– The Enrollee should set "lec" Property to 6, if the Auth type is not supported by the Enrollee.

– The Enrollee should set "lec" Property to 7, if the Encryption type is not supported by the Enrollee.

– The Enrollee should set "lec" Property to 8, if it fails to connect due to wrong Auth type information (even though it's supported by the Enrollee).

347   –   The Enrollee should set "lec" Property to 9, if it fails to connect due to wrong Encryption type
348   information (even though it's supported by the Enrollee).

349 When using Wi-Fi as the connectivity between the Enrollee and Mediator, if the Enrollee fails to
350 connect to the Enroller, it shall again make itself discoverable as a Soft AP (in case it destroyed
351 its Soft AP earlier).

352 **8.5 Example Easy Setup Flow**

353 Figure 3 shows an example Easy Setup flow for informative purposes:
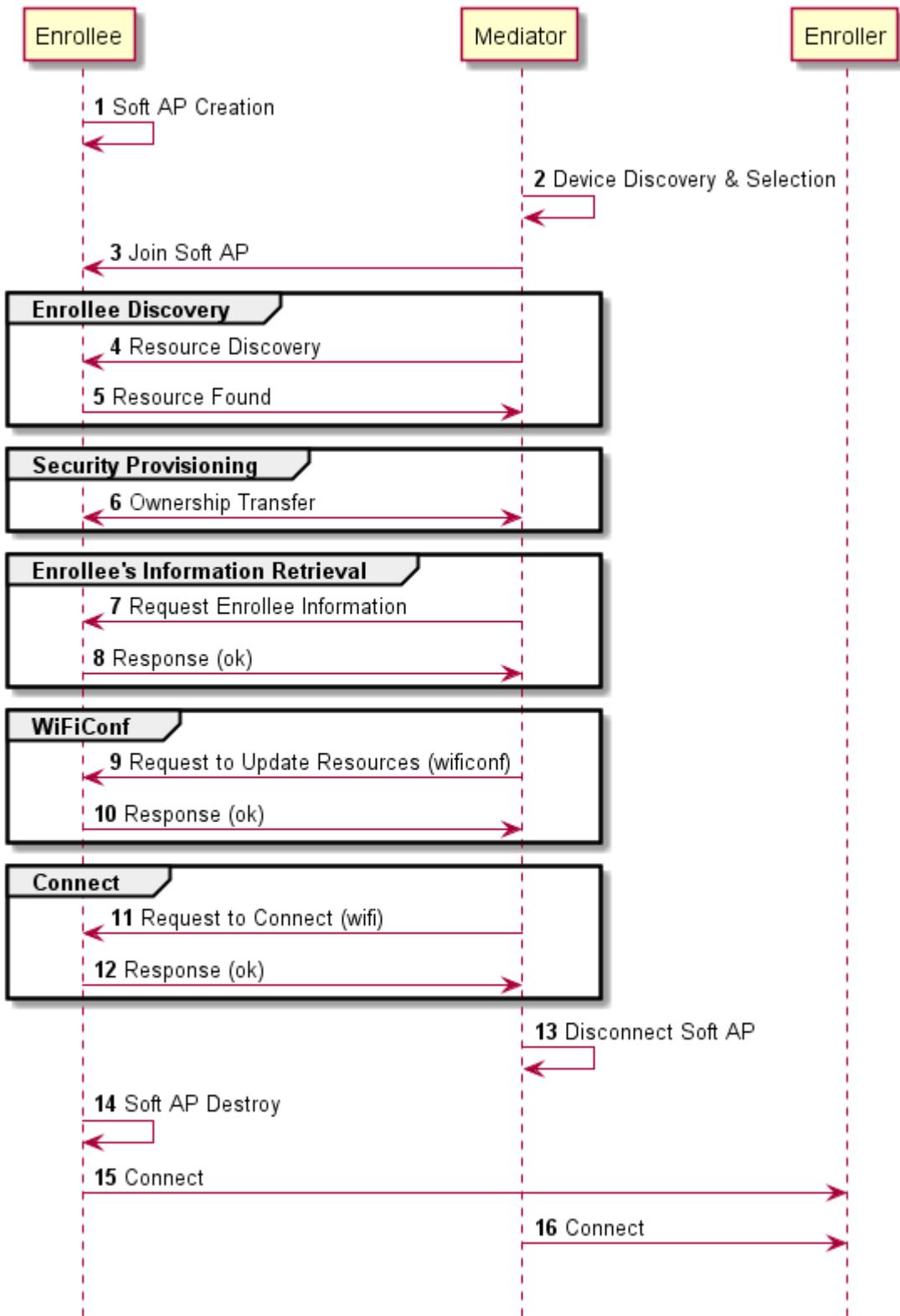
Figure 3 – Easy Setup Flow (Informative)

357 The example flow in Figure 1Figure 3 undergoes security provisioning (step 6) during Easy Setup.
358 Alternatively, security provisioning can be done before Enrollee Discovery (steps 4 and 5) if
359 preferred. Please refer to the ISO/IEC 30118-2:2018 for more information on the different scenarios.

8.6 Easy Setup SSID Tags

361 If using Wi-Fi as the connectivity between the Enrollee and the Mediator then the Enrollee's Soft
362 AP SSID should contain exactly one of the following Easy Setup SSID tags:

363 – "OCF_"

364   – Prefix tag that has to be at the beginning of the SSID.

365   – Example: OCF_MySSID

366 – "_OCF"

367   – Suffix tag that has to be at the end of the SSID.

368   – Example: MySSID_OCF

369 These tags are case sensitive.

8.7 Easy Setup Information Element

8.7.1 Overview

372 If using Wi-Fi as the connectivity between the Enrollee and the Mediator then the Enrollee's Soft
373 AP beacon should contain the Easy Setup Information Element. The information element provides
374 additional information about the device such as a friendly name or device manufacturer for the
375 mediator application. The mediator application can then use this information to provide a better
376 user experience.

8.7.2 OCF Device Information Element (IE)

378 The Easy Setup Information Element has the structure shown in Figure 4

1 byte	1 byte	3 bytes	1 byte	<252 bytes
Type = 221	Length	CID = 6A 40 65	OCF IE Type = 0	Data

379 **Figure 4 – Easy Setup Information Element Definition**

380 – Type is a unique id allocated by the IEEE registrar to identify different information elements
381   from each other. The Easy Setup Information Element shall have a Type value of 221 which is
382   standard vendor specific information element.

383 – Length shall indicate the total size of CID, OCF IE Type, and Data in bytes.

384 – Company ID (CID) is a unique 24-bit identifier for a specific company or organization. The Easy
385   Setup Information Element shall have a CID value of 6A 40 65.

386 – OCF IE Type is the identifier of the specific IE within OCF. The OCF IE Type shall be set to 0
387   for Easy Setup.

388 – Data is a set of type-length-value (TLV) structures that represent the device information in Table
389   1. The length of this field shall be less than 252 bytes.

390

391 Each TLV has the structure shown in Figure 5.

1 byte	1 byte	<250 bytes
Type	Length	Value

392   **Figure 5 – Type-Length-Value Structure**

393   – Type shall indicate the type of the field from Table 7.

394   – Length shall indicate the length of the Value in bytes.

395   – Value shall represent the corresponding information for specific TLV type from Table 7.

396   Data is a set of TLVs as defined in Table 7.

397   **Table 7 – Easy Setup Information Element TLVs**

Type	Length (bytes)	Value	Description of TLV	# of Occurrences in IE or IEC	Required
1	<65	Friendly name of the device	Device Friendly Name	1	Y
2	<27	Device Type	Device type/Class	>=1	Y
3	<65	Name of Device Manufacturer	Manufacturer Name	1	Y
4	<43	Language tag for strings	See IETF RFC 5646	1	Y
5	16	Protocol Independent ID in network byte order	See ISO/IEC 30118-1:2018	1	Y
101	<65	Device Type/Class	Device Type as string	>=0	N

398

399   The TLVs may be set in any order inside an IE or IEC. All strings shall be UTF-8 encoded and shall
400   not include a null terminator. All TLVs in Table 7 with a required value of "Y" shall be included in
401   the IE or IEC (if multiple IEs are required). The value of each TLV shall meet the length
402   requirements specified in Table 1.

403   **8.7.2.1   Device Friendly Name (Type 1)**

404   User readable string representing the friendly name of the device that is beaconing and ready to
405   undergo Easy Setup. This should match "n" from "oic.wk.d" as defined in the ISO/IEC 30118-
406   1:2018.

407   This string is in the same language specified in the type 4 TLV.

408   **8.7.2.2   Device Type (Type 2)**

409   Device type shall be the shortened form of Device Type as specified in the ISO/IEC 30118-5:2018. For
410   example:
411

412   – Device Type as specified in the ISO/IEC 30118-5:2018: "oic.d.airconditioner"

413   – Device Type as specified in a type 2 TLV: "airconditioner"

414   In cases where the device supports multiple functions, several type 2 TLVs may be included to
415   represent each function of the device.

If the device does not support any of the functions as specified in the ISO/IEC 30118-5:2018, at least one type 101 TLV shall be included. Type 101 TLV contains a user readable string in the same language specified in the type 4 TLV. (Ex: "Lock").

If the device supports more than one function, a mix of type 2 and type 101 TLVs may be used depending on which functions are defined in the ISO/IEC 30118-5:2018.

8.7.2.3 Device Manufacturer Name (Type 3)

User readable string representing the manufacturer name of the device that is beaconing and ready to undergo Easy Setup. This should match "mnmn" Property from "oic.wk.p" as defined in the ISO/IEC 30118-1:2018.

This string is in the same language specified in the type 4 TLV.

8.7.2.4 Language Tag (Type 4)

The language of all strings shall be specified in a type 4 TLV. The value of the type 4 TLV shall contain a language tag as described in IETF RFC 5646 (Ex: "en-us"). If the actual length of the language tag exceeds 42 bytes the manufacturer shall exclude subtags on the language tag until it is less than 43 bytes.

Please see 8.7.2.8 for information on supporting multiple languages.

If an IE contains a TLV that is a string (i.e. type 1, type 3 or type 101), then a type 4 TLV corresponding to the language of the string(s) shall also be present in the IE.

8.7.2.5 Protocol Independent ID (Type 5)

This shall match "piid" from "oic.wk.d" as defined in the ISO/IEC 30118-1:2018.

The piid in the TLV shall be in network byte order.

8.7.2.6 Multiple Information Elements

Additional Easy Setup IEs may be present in the Soft AP beacon in the following situations:

– The total size of the TLVs is larger than the size of Data as defined in an Easy Setup Information Element.
– Support for multiple languages is necessary.

Two or more Easy Setup Information Elements are referred to as an Information Element Collection (IEC).

8.7.2.7 IEC for Large TLV Size Support

If a TLV or set of TLVs will not fit into the current IE, a manufacturer may add additional Easy Setup IEs to contain the TLV/s thereby creating or extending an IEC. The additional IE shall contain the following fields as described in 8.7.2:

– Type
– Length
– CID
– OCF IE Type

If an IE contains a TLV that is a string (i.e. type 1, type 3 or type 101), then a type 4 TLV corresponding to the language of the string(s) shall also be present in the IE.

8.7.2.8 IEC for Multiple Language Support

A manufacturer may include additional Easy Setup IEs to support multiple languages in the Soft AP beacon. In the case that a manufacturer needs to provide device information in more than one language, they shall include an additional copy of the IE/IEC for each additional language. Each additional IE/IEC shall include all of the mandatory TLVs defined in 8.7.2.

9   Security

A Device shall meet the Wi-Fi Easy Setup security requirements specified in ISO/IEC 30118-2:2018.

Annex A(normative)

462

463 **OpenAPI 2.0 specification definitions**

A.1 List of Resource Type definitions

465 Table A.1 contains the list of defined resources in this document.

466 **Table A.1 – Alphabetized list of resources**

Friendly Name (informative)		Resource Type (rt)	Clause
Device Configuration		"oic.r.devconf"	A.2
Easy Setup		"oic.r.easysetup"	A.3
Wi-Fi Configuration		"oic.r.wificonf"	A.4

467

A.2 Device Configuration

A.2.1   Introduction

470 The Device configuration Resource stores Device settings such as the Device name. Vendor-
471 specific   information   can   be   added   to   the   Resource.
472 The Device name is a human-friendly name read by a Mediator during easy setup.
473

A.2.2   Example URI

475 /example/DevConfResURI

A.2.3   Resource type

477 The Resource Type is defined as: "oic.r.devconf".

A.2.4   OpenAPI 2.0 definition

```
479  {
480    "swagger": "2.0",
481    "info": {
482      "title": "Device Configuration",
483      "version": "20190306",
484      "license": {
485        "name": "OCF Data Model License",
486        "url":
487  "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
488  CENSE.md",
489        "x-copyright": "Copyright 2018-2019 Open Connectivity Foundation, Inc. All rights reserved."
490      },
491      "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
492    },
493    "schemes": ["http"],
494    "consumes": ["application/json"],
495    "produces": ["application/json"],
496    "paths": {
497      "/example/DevConfResURI" : {
498        "get": {
499          "description": "The Device configuration Resource stores Device settings such as the Device
500  name. Vendor-specific information can be added to the Resource.\nThe Device name is a human-friendly
501  name read by a Mediator during easy setup.\n",
502          "parameters": [
503            {"$ref": "#/parameters/interface"}
```

```
504              ],
505              "responses": {
506                "200": {
507                  "description" : "",
508                  "x-example": {
509                    "rt": ["oic.r.devconf"],
510                    "dn" : "My Refrigerator"
511                  },
512                  "schema": { "$ref": "#/definitions/DevConf" }
513                }
514              }
515            }
516          }
517        },
518        "parameters": {
519          "interface" : {
520            "in" : "query",
521            "name" : "if",
522            "type" : "string",
523            "enum" : ["oic.if.r", "oic.if.baseline"]
524          }
525        },
526        "definitions": {
527          "DevConf" : {
528            "properties": {
529              "rt" : {
530                "description": "Resource Type of the Resource",
531                "items": {
532                  "enum": ["oic.r.devconf"],
533                  "maxLength": 64,
534                  "type": "string"
535                },
536                "minItems": 1,
537                "readOnly": true,
538                "uniqueItems": true,
539                "type": "array"
540              },
541              "n" : {
542                "$ref":
543    "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
544    schema.json#/definitions/n"
545              },
546              "id" : {
547                "$ref":
548    "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
549    schema.json#/definitions/id"
550              },
551              "if" : {
552                "description": "The OCF Interfaces supported by this Resource",
553                "items": {
554                  "enum": [
555                    "oic.if.r",
556                    "oic.if.baseline"
557                  ],
558                  "type": "string",
559                  "maxLength": 64
560                },
561                "minItems": 2,
562                "readOnly": true,
563                "uniqueItems": true,
564                "type": "array"
565              },
566              "dn": {
567                "oneOf": [
568                  {
569                    "type": "string",
570                    "description": "Indicates a pre-configured Device name in language indicated by 'dl'
571    in /oic/con;  presented by an Enrollee Device to a Mediator Device during the easy-setup process",
572                    "pattern": "^.*$",
573                    "readOnly": true
```

```
574                    },
575                    {
576                      "type": "array",
577                      "items": {
578                        "type": "object",
579                        "properties": {
580                          "language": {
581                            "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
582    schema.json#/definitions/language-tag",
583                            "readOnly": true,
584                            "description": "An RFC 5646 language tag."
585                          },
586                          "value": {
587                            "type": "string",
588                            "description": "Pre-configured Device name in the indicated language.",
589                            "pattern": "^.*$",
590                            "readOnly": true
591                          }
592                        }
593                      },
594                      "minItems" : 1,
595                      "readOnly": true,
596                      "description": "Localized device name."
597                    }
598                  ]
599                }
600              },
601              "type" : "object",
602              "required": ["dn"]
603          }
604        }
605    }
606
```

A.2.5   Property definition

Table A.2 defines the Properties that are part of the "oic.r.devconf" Resource Type.

Table A.2 – The Property definitions of the Resource with type "rt" = "oic.r.devconf".

Property name	Value type	Mandatory	Access mode	Description
id	multiple types: see schema	No	Read Write	
n	multiple types: see schema	No	Read Write	
dn	multiple types: see schema	Yes	Read Write	
rt	array: see schema	No	Read Only	Resource Type of the Resource.
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.

A.2.6   CRUDN behaviour

Table A.3 defines the CRUDN operations that are supported on the "oic.r.devconf" Resource Type.

Table A.3 – The CRUDN operations of the Resource with type "rt" = "oic.r.devconf".

Create	Read	Update	Delete	Notify
	get			observe

613   ## A.3   Easy Setup Collection

614   ### A.3.1   Introduction

615   The Easy Setup Resource stores useful information including the current status of unboxing a
616   Device and the last error code which are produced in the process of easy setup.
617   Note that the Easy Setup Resource is a Collection Resource, which contains Links to WiFiConf,
618   and DevConf Resources and may additionally contain Links to other Resources.
619

620   ### A.3.2   Example URI

621   /EasySetupResURI

622   ### A.3.3   Resource type

623   The Resource Type is defined as: "oic.r.easysetup, oic.wk.col".

624   ### A.3.4   OpenAPI 2.0 definition

```
625   {
626     "swagger": "2.0",
627     "info": {
628       "title": "Easy Setup Collection",
629       "version": "20190327",
630       "license": {
631         "name": "OCF Data Model License",
632         "url":
633   "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
634   CENSE.md",
635         "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
636       },
637       "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
638     },
639     "schemes": ["http"],
640     "consumes": ["application/json"],
641     "produces": ["application/json"],
642     "paths": {
643       "/EasySetupResURI?if=oic.if.ll" : {
644         "get": {
645           "description": "The Easy Setup Resource stores useful information including the current
646   status of unboxing a Device and the last error code which are produced in the process of easy
647   setup.\nNote that the Easy Setup Resource is a Collection Resource, which contains Links to
648   WiFiConf, and DevConf Resources and may additionally contain Links to other Resources.\n",
649           "parameters": [
650             {"$ref": "#/parameters/interface-all"}
651           ],
652           "responses": {
653             "200": {
654               "description" : "",
655               "x-example":
656                 [
657                   {
658                     "href": "/EasySetupResURI",
659                     "rt": ["oic.r.easysetup", "oic.wk.col"],
660                     "if": ["oic.if.b"],
661                     "p":{"bm":3},
662                     "eps": [
663                       {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
664                     ],
665                     "rel":["self", "item"]
666                   },
667                   {
668                     "href": "/WiFiConfResURI",
669                     "rt":    ["oic.r.wificonf"],
670                     "if":    ["oic.if.baseline"],
671                     "p":{"bm":3},
672                     "eps": [
```

```
673                          {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
674                        ]
675                      },
676                      {
677                        "href": "/DevConfResURI",
678                        "rt":   ["oic.r.devconf"],
679                        "if":   ["oic.if.baseline"],
680                        "p":{"bm":3},
681                        "eps": [
682                          {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
683                        ]
684                      }
685                    ],
686                "schema": { "$ref": "#/definitions/slinks" }
687              }
688            }
689          }
690        },
691      "/EasySetupResURI?if=oic.if.b" : {
692        "get": {
693          "description": "The Easy Setup Resource stores useful information including the current
694    status of unboxing a Device and the last error code which are produced in the process of easy
695    setup.\nNote that the Easy Setup Resource is a Collection Resource, which contains Links to
696    WiFiConf, and DevConf Resources and may additionally contain Links to other Resources.\n",
697          "parameters": [
698            {"$ref": "#/parameters/interface-all"}
699          ],
700          "responses": {
701            "200": {
702              "description" : "",
703              "x-example":
704                [
705                  {
706                    "href": "/EasySetupResURI",
707                    "rep":{
708                      "ps" : 0,
709                      "lec": 0,
710                      "cn": [1]
711                    }
712                  },
713                  {
714                    "href": "/WiFiConfResURI",
715                    "rep":{
716                      "swmt" : ["A", "B", "G"],
717                      "swf": ["2.4G", "5G"],
718                      "tnn": "Home_AP_SSID",
719                      "cd": "Home_AP_PWD",
720                      "wat": "WPA2_PSK",
721                      "wet": "AES",
722                      "swat": ["WPA_PSK", "WPA2_PSK"],
723                      "swet": ["TKIP", "AES", "TKIP_AES"]
724                    }
725                  },
726                  {
727                    "href": "/DevConfResURI",
728                    "rep":{
729                      "dn" : "My Refrigerator"
730                    }
731                  }
732                ],
733              "schema": { "$ref": "#/definitions/sbatch" }
734            }
735          }
736        },
737        "post": {
738          "description": "Able to deliver Wi-Fi, Device configuration and other
739    configuration\ninformation in a batch by utilizing 'batch' OCF Interface.\nIf you want to deliver
740    Wi-Fi and Device configuration information in a batch,\nyou can write all Properties you want to
741    send with a 'batch' OCF Interface.\nThe below example is the case to send Easy Setup and Wi-Fi
742    configuration\n(i.e. connection type, target network, auth type information) in a batch.\n",
```

```
743              "parameters": [
744                {"$ref": "#/parameters/interface-update"},
745                {
746                  "name": "body",
747                  "in": "body",
748                  "required": true,
749                  "schema": { "$ref": "#/definitions/sbatch-update" },
750                  "x-example":
751                    [
752                      {
753                        "href": "/EasySetupResURI",
754                        "rep":{
755                          "cn": [1]
756                        }
757                      },
758                      {
759                        "href": "/WiFiConfResURI",
760                        "rep":{
761                          "tnn": "Home_AP_SSID",
762                          "cd": "Home_AP_PWD",
763                          "wat": "WPA2_PSK",
764                          "wet": "AES"
765                        }
766                      }
767                    ]
768                }
769              ],
770              "responses": {
771                "200": {
772                  "description" : "",
773                  "x-example":
774                    [
775                      {
776                        "href": "/EasySetupResURI",
777                        "rep" : {
778                          "ps" : 0,
779                          "lec": 0,
780                          "cn": [1]
781                        }
782                      },
783                      {
784                        "href": "/WiFiConfResURI",
785                        "rep" : {
786                          "swmt" : ["A", "B", "G"],
787                          "swf": ["2.4G", "5G"],
788                          "tnn": "Home_AP_SSID",
789                          "cd": "Home_AP_PWD",
790                          "wat": "WPA2_PSK",
791                          "wet": "AES",
792                          "swat": ["WPA_PSK", "WPA2_PSK"],
793                          "swet": ["TKIP", "AES", "TKIP_AES"]
794                        }
795                      },
796                      {
797                        "href": "/DevConfResURI",
798                        "rep" : {
799                          "dn" : "My Refrigerator"
800                        }
801                      }
802                    ],
803                  "schema": { "$ref": "#/definitions/sbatch" }
804                }
805              }
806            }
807          },
808          "/EasySetupResURI?if=oic.if.baseline" : {
809            "get": {
810              "description": "The Easy Setup Resource stores useful information including the current
811    status of unboxing a Device and the last error code which are produced in the process of easy
812    setup.\nNote that the Easy Setup Resource is a Collection Resource, which contains Links to
```

```
813    WiFiConf, and DevConf Resources and may additionally contain Links to other Resources.\n",
814            "parameters": [
815              {"$ref": "#/parameters/interface-all"}
816            ],
817            "responses": {
818              "200": {
819                "description" : "",
820                "x-example":
821                  {
822                    "rt" : ["oic.r.easysetup", "oic.wk.col"],
823                    "if" : ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
824                    "ps" : 0,
825                    "lec": 0,
826                    "cn": [1],
827                    "links": [
828                      {
829                        "href": "/EasySetupResURI",
830                        "rt": ["oic.r.easysetup", "oic.wk.col"],
831                        "if": ["oic.if.b"],
832                        "p":{"bm":3},
833                        "eps": [
834                          {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
835                        ],
836                        "rel":["self", "item"]
837                      },
838                      {
839                        "href": "/WiFiConfResURI",
840                        "rt":   ["oic.r.wificonf"],
841                        "if":   ["oic.if.baseline"],
842                        "p":{"bm":3},
843                        "eps": [
844                          {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
845                        ]
846                      },
847                      {
848                        "href": "/DevConfResURI",
849                        "rt":   ["oic.r.devconf"],
850                        "if":   ["oic.if.baseline"],
851                        "p":{"bm":3},
852                        "eps": [
853                          {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
854                        ]
855                      }
856                    ]
857                  },
858                "schema": { "$ref": "#/definitions/EasySetup" }
859              }
860            }
861          },
862          "post": {
863            "description": "Able to update connection type to attempt to connect to the Enroller to
864    start during while posting to /EasySetupResURI\nThe below example is the case to send Easy Setup
865    configuration\n(i.e. connection type) in a post.\n",
866            "parameters": [
867              {"$ref": "#/parameters/interface-update"},
868              {
869                "name": "body",
870                "in": "body",
871                "required": true,
872                "schema": { "$ref": "#/definitions/EasySetupUpdate" },
873                "x-example":
874                  {
875                    "cn": [1]
876                  }
877              }
878            ],
879            "responses": {
880              "200": {
881                "description" : "",
882                "x-example":
```

```
883                    {
884                      "rt" : ["oic.r.easysetup", "oic.wk.col"],
885                      "if" : ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
886                      "ps" : 0,
887                      "lec": 0,
888                      "cn": [1],
889                      "links": [
890                        {
891                          "href": "/EasySetupResURI",
892                          "rt": ["oic.r.easysetup", "oic.wk.col"],
893                          "if": ["oic.if.b", "oic.if.ll", "oic.if.baseline"],
894                          "p":{"bm":3},
895                          "eps": [
896                            {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
897                          ],
898                          "rel":["self", "item"]
899                        },
900                        {
901                          "href": "/WiFiConfResURI",
902                          "rt":    ["oic.r.wificonf"],
903                          "if":    ["oic.if.rw", "oic.if.baseline"],
904                          "p":{"bm":3},
905                          "eps": [
906                            {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
907                          ]
908                        },
909                        {
910                          "href": "/DevConfResURI",
911                          "rt":    ["oic.r.devconf"],
912                          "if":    ["oic.if.r", "oic.if.baseline"],
913                          "p":{"bm":3},
914                          "eps": [
915                            {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
916                          ]
917                        }
918                      ]
919                    },
920                    "schema": { "$ref": "#/definitions/EasySetup" }
921                  }
922                }
923              }
924            }
925          },
926          "parameters": {
927            "interface-all" : {
928              "in" : "query",
929              "name" : "if",
930              "type" : "string",
931              "enum" : ["oic.if.ll","oic.if.b","oic.if.baseline"]
932            },
933            "interface-update" : {
934              "in" : "query",
935              "name" : "if",
936              "type" : "string",
937              "enum" : ["oic.if.b", "oic.if.baseline"]
938            }
939          },
940          "definitions": {
941            "oic.oic-link": {
942              "type": "object",
943              "properties": {
944                "anchor": {
945                  "$ref":
946    "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
947    schema.json#/definitions/anchor"
948                },
949                "di": {
950                  "$ref":
951    "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
952    schema.json#/definitions/di"
```

```
953              },
954              "eps": {
955                  "$ref":
956   "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
957   schema.json#/definitions/eps"
958              },
959              "href": {
960                  "$ref":
961   "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
962   schema.json#/definitions/href"
963              },
964              "ins": {
965                  "$ref":
966   "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
967   schema.json#/definitions/ins"
968              },
969              "p": {
970                  "$ref":
971   "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
972   schema.json#/definitions/p"
973              },
974              "rel": {
975                  "$ref":
976   "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
977   schema.json#/definitions/rel_array"
978              },
979              "title": {
980                  "$ref":
981   "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
982   schema.json#/definitions/title"
983              },
984              "type": {
985                  "$ref":
986   "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
987   schema.json#/definitions/type"
988              },
989              "if": {
990                  "description": "The OCF Interfaces supported by the target Resource",
991                  "items": {
992                      "enum": [
993                          "oic.if.baseline",
994                          "oic.if.ll",
995                          "oic.if.b",
996                          "oic.if.r",
997                          "oic.if.rw"
998                      ],
999                      "type": "string",
1000                     "maxLength": 64
1001                 },
1002                 "minItems": 1,
1003                 "uniqueItems": true,
1004                 "type": "array"
1005             },
1006             "rt": {
1007                 "description": "Resource Type of the target Resource",
1008                 "items": {
1009                     "maxLength": 64,
1010                     "type": "string"
1011                 },
1012                 "minItems": 1,
1013                 "uniqueItems": true,
1014                 "type": "array"
1015             }
1016         },
1017         "required": [
1018             "href",
1019             "rt",
1020             "if"
1021         ]
1022     },
```

```
1023          "slinks" : {
1024            "type": "array",
1025            "items": {
1026              "$ref": "#/definitions/oic.oic-link"
1027            }
1028          },
1029          "sbatch" : {
1030            "minItems" : 1,
1031            "items" : {
1032              "additionalProperties": true,
1033              "properties": {
1034                "href": {
1035                  "$ref":
1036     "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1037     schema.json#/definitions/href"
1038                },
1039                "rep": {
1040                  "description": "The response payload from a single Resource",
1041                  "type": "object",
1042                  "anyOf": [
1043                    {
1044                        "$ref": "#/definitions/EasySetup"
1045                    },
1046                    {
1047                        "$ref": "https://openconnectivityfoundation.github.io/core-
1048     extensions/swagger2.0/oic.r.wificonf.swagger.json#/definitions/WiFiConf"
1049                    },
1050                    {
1051                        "$ref": "https://openconnectivityfoundation.github.io/core-
1052     extensions/swagger2.0/oic.r.devconf.swagger.json#/definitions/DevConf"
1053                    }
1054                  ]
1055                }
1056              },
1057              "required": [
1058                "href",
1059                "rep"
1060              ],
1061              "type": "object"
1062            },
1063            "type" : "array"
1064          },
1065          "sbatch-update" : {
1066            "minItems" : 1,
1067            "items" : {
1068              "additionalProperties": true,
1069              "description": "Array of Resource representations to apply to the batch Collection, using
1070     href to indicate which resource(s) in the batch to update. If the href Property is empty,
1071     effectively making the URI reference to the Collection itself, the representation is to be applied
1072     to all Resources in the batch",
1073              "properties": {
1074                "href": {
1075                  "$ref":
1076     "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1077     schema.json#/definitions/href"
1078                },
1079                "rep": {
1080                  "description": "The response payload from a single Resource",
1081                  "type": "object",
1082                  "anyOf": [
1083                    {
1084                      "$ref": "#/definitions/EasySetupUpdate"
1085                    },
1086                    {
1087                        "$ref": "https://openconnectivityfoundation.github.io/core-
1088     extensions/swagger2.0/oic.r.wificonf.swagger.json#/definitions/WiFiConfUpdate"
1089                    }
1090                  ]
1091                }
1092              },
```

```
1093            "required": [
1094              "href",
1095              "rep"
1096            ],
1097          "type": "object"
1098        },
1099        "type" : "array"
1100      },
1101      "EasySetup" : {
1102        "properties": {
1103          "n" : {
1104            "$ref":
1105  "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
1106  schema.json#/definitions/n"
1107          },
1108          "rts" : {
1109            "description": "Resource Type of the Resources within the Collection",
1110            "items": {
1111              "maxLength": 64,
1112              "type": "string"
1113            },
1114            "minItems": 1,
1115            "uniqueItems": true,
1116            "readOnly": true,
1117            "type": "array"
1118          },
1119          "id" : {
1120            "$ref":
1121  "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
1122  schema.json#/definitions/id"
1123          },
1124          "rts-m" : {
1125            "description": "Resource Type of the mandatory Resources within the Collection",
1126            "items": {
1127              "maxLength": 64,
1128              "type": "string"
1129            },
1130            "minItems": 1,
1131            "uniqueItems": true,
1132            "readOnly": true,
1133            "type": "array"
1134          },
1135          "if" : {
1136            "description": "The OCF Interfaces supported by this Resource",
1137            "items": {
1138              "enum": [
1139                "oic.if.ll",
1140                "oic.if.baseline",
1141                "oic.if.b"
1142              ],
1143              "type": "string",
1144              "maxLength": 64
1145            },
1146            "minItems": 2,
1147            "uniqueItems": true,
1148            "readOnly": true,
1149            "type": "array"
1150          },
1151          "rt" : {
1152            "items": {
1153              "enum": [
1154                "oic.r.easysetup",
1155                "oic.wk.col"
1156              ],
1157              "type": "string",
1158              "maxLength": 64
1159            },
1160            "minItems": 2,
1161            "type": "array",
1162            "uniqueItems": true
```

```
1163            },
1164            "ps" : {
1165               "description": "Indicates the easy setup status of the Device. (0: Need to Setup, 1:
1166    Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller, 4~254: Reserved,
1167    255: EOF)",
1168               "enum": [
1169                  0,
1170                  1,
1171                  2,
1172                  3
1173               ],
1174               "readOnly": true,
1175               "type": "integer"
1176            },
1177            "lec" : {
1178               "description": "Indicates a failure reason (0: NO error, 1: A given SSID is not found, 2:
1179    Wi-Fi's password is wrong, 3: IP address is not allocated, 4: No internet connection, 5: Timeout, 6:
1180    Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not supported by the
1181    Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller), 9: Wi-Fi
1182    Encryption Type is wrong (failure while connecting to the Enroller), 10~254: Reserved, 255: Unknown
1183    error)",
1184               "enum": [
1185                  0,
1186                  1,
1187                  2,
1188                  3,
1189                  4,
1190                  5,
1191                  6,
1192                  7,
1193                  8,
1194                  9,
1195                  255
1196               ],
1197               "readOnly": true,
1198               "type": "integer"
1199            },
1200            "cn" : {
1201               "description": "Indicates an array of connection types that trigger an attempt to connect
1202    to the Enroller to start.",
1203               "items": {
1204                  "description": "Connection type to attempt. (1 : Wi-Fi, 2 : other entities / transports
1205    to be added in future (e.g. Connect to cloud / BLE))",
1206                  "type": "integer"
1207               },
1208               "type": "array"
1209            },
1210            "links" : {
1211               "type": "array",
1212               "description": "A set of OCF Links.",
1213               "items": {
1214                  "$ref": "#/definitions/oic.oic-link"
1215               }
1216            }
1217         },
1218         "type" : "object",
1219         "required": ["ps","lec","cn"]
1220      },
1221      "EasySetupUpdate" : {
1222         "additionalProperties": true,
1223         "description": "Update to writeable values in EasySetupResURI",
1224         "properties": {
1225            "cn" : {
1226               "description": "Indicates an array of connection types that trigger an attempt to connect
1227    to the Enroller to start.",
1228               "items": {
1229                  "description": "Connection type to attempt. (1 : Wi-Fi, 2 : other entities / transports
1230    to be added in future (e.g. Connect to cloud / BLE))",
1231                  "type": "integer"
1232               },
```

```
1233              "type": "array"
1234            }
1235          },
1236          "required": [
1237            "cn"
1238          ],
1239          "type": "object"
1240        }
1241      }
1242  }
1243
```

A.3.5   Property definition

Table A.4 defines the Properties that are part of the "oic.r.easysetup, oic.wk.col" Resource Type.

Table A.4 – The Property definitions of the Resource with type "rt" = "oic.r.easysetup, oic.wk.col".

Property name	Value type	Mandatory	Access mode	Description
rep	object: see schema	Yes	Read Write	The response payload from a single Resource.
href	multiple types: see schema	Yes	Read Write	
rep	object: see schema	Yes	Read Write	The response payload from a single Resource.
href	multiple types: see schema	Yes	Read Write	
links	array: see schema	No	Read Write	A set of OCF Links.
rts-m	array: see schema	No	Read Only	Resource Type of the mandatory Resources within the Collection.
n	multiple types: see schema	No	Read Write	
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.
ps	integer	Yes	Read Only	Indicates the easy setup status of the Device. (0: Need to Setup, 1: Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller, 4~254: Reserved, 255: EOF).
lec	integer	Yes	Read Only	Indicates a failure reason (0: NO error, 1: A given SSID is not found, 2: Wi-Fi's password is wrong, 3: IP address is not allocated, 4: No internet connection, 5: Timeout, 6: Wi-Fi Auth Type is not supported by the

29

				Enrollee, 7: Wi-Fi Encryption Type is not supported by the Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller), 9: Wi-Fi Encryption Type is wrong (failure while connecting to the Enroller), 10~254: Reserved, 255: Unknown error).
rt	array: see schema	No	Read Write	
rts	array: see schema	No	Read Only	Resource Type of the Resources within the Collection.
cn	array: see schema	Yes	Read Write	Indicates an array of connection types that trigger an attempt to connect to the Enroller to start.
id	multiple types: see schema	No	Read Write	
rt	array: see schema	Yes	Read Write	Resource Type of the target Resource.
href	multiple types: see schema	Yes	Read Write	
if	array: see schema	Yes	Read Write	The OCF Interfaces supported by the target Resource.
type	multiple types: see schema	No	Read Write	
p	multiple types: see schema	No	Read Write	
ins	multiple types: see schema	No	Read Write	
title	multiple types: see schema	No	Read Write	
anchor	multiple types: see schema	No	Read Write	
rel	multiple types: see schema	No	Read Write	
eps	multiple types: see schema	No	Read Write	
di	multiple types: see schema	No	Read Write	
cn	array: see schema	Yes	Read Write	Indicates an array of connection types that trigger an attempt to connect to the Enroller to start.

A.3.6   CRUDN behaviour

1249   Table A.5defines the CRUDN operations that are supported on the "oic.r.easysetup, oic.wk.col"
1250   Resource Type.

1251   **Table A.5 – The CRUDN operations of the Resource with type "rt" = "oic.r.easysetup,**
1252   **oic.wk.col".**

Create	Read	Update	Delete	Notify
	get	post		observe

1253   **A.4   Wi-Fi Configuration**

1254   **A.4.1   Introduction**

1255   WiFiConf   Resource   stores   essential   information   to   help   an   unboxing   Device
1256   to   connect   to   an   existing   Wi-Fi   AP.
1257

1258   **A.4.2   Example URI**

1259   /WiFiConfResURI

1260   **A.4.3   Resource type**

1261   The Resource Type is defined as: "oic.r.wificonf".

1262   **A.4.4   OpenAPI 2.0 definition**

```
1263    {
1264      "swagger": "2.0",
1265      "info": {
1266        "title": "Wi-Fi Configuration",
1267        "version": "20190327",
1268        "license": {
1269          "name": "OCF Data Model License",
1270          "url":
1271    "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
1272    CENSE.md",
1273          "x-copyright": "Copyright 2018-2019 Open Connectivity Foundation, Inc. All rights reserved."
1274        },
1275        "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
1276      },
1277      "schemes": ["http"],
1278      "consumes": ["application/json"],
1279      "produces": ["application/json"],
1280      "paths": {
1281        "/WiFiConfResURI?if=oic.if.rw" : {
1282          "get": {
1283            "description": "The WiFiConf Resource stores essential information to help an unboxing
1284    Device connect to an existing Wi-Fi AP.\n",
1285            "parameters": [
1286              {"$ref": "#/parameters/interface-all"}
1287            ],
1288            "responses": {
1289              "200": {
1290                "description" : "",
1291                "x-example":
1292                  {
1293                    "tnn": "Home_AP_SSID",
1294                    "swmt" : ["A", "B", "G"],
1295                    "swf": ["2.4G", "5G"],
1296                    "cd": "Home_AP_PWD",
1297                    "wat": "WPA2_PSK",
1298                    "wet": "AES",
1299                    "swat": ["WPA_PSK", "WPA2_PSK"],
```

```
1300                    "swet": ["TKIP", "AES", "TKIP_AES"]
1301                  },
1302                "schema": { "$ref": "#/definitions/WiFiConf" }
1303              }
1304            }
1305          },
1306          "post": {
1307            "description": "Deliver Wi-Fi AP's information for an unboxing Device to connect to it.\n",
1308            "parameters": [
1309              {"$ref": "#/parameters/interface-all"},
1310              {
1311                "name": "body",
1312                "in": "body",
1313                "required": true,
1314                "schema": { "$ref": "#/definitions/WiFiConfUpdate" },
1315                "x-example":
1316                  {
1317                    "tnn": "Home_AP_SSID",
1318                    "cd": "Home_AP_PWD",
1319                    "wat": "WPA2_PSK",
1320                    "wet": "AES"
1321                  }
1322              }
1323            ],
1324            "responses": {
1325              "200": {
1326                "description" : "",
1327                "x-example":
1328                  {
1329                    "tnn": "Home_AP_SSID",
1330                    "swmt" : ["A", "B", "G"],
1331                    "swf": ["2.4G", "5G"],
1332                    "cd": "Home_AP_PWD",
1333                    "wat": "WPA2_PSK",
1334                    "wet": "AES",
1335                    "swat": ["WPA_PSK", "WPA2_PSK"],
1336                    "swet": ["TKIP", "AES", "TKIP_AES"]
1337                  },
1338                "schema": { "$ref": "#/definitions/WiFiConf" }
1339              }
1340            }
1341          }
1342        },
1343        "/WiFiConfResURI?if=oic.if.baseline" : {
1344          "get": {
1345            "description": "WiFiConf Resource stores essential information to help an unboxing
1346    Device\nto connect to an existing Wi-Fi AP.\n",
1347            "parameters": [
1348              {"$ref": "#/parameters/interface-all"}
1349            ],
1350            "responses": {
1351              "200": {
1352                "description" : "",
1353                "x-example":
1354                  {
1355                    "rt": ["oic.r.wificonf"],
1356                    "if": ["oic.if.rw", "oic.if.baseline"],
1357                    "swmt" : ["A", "B", "G"],
1358                    "swf": ["2.4G", "5G"],
1359                    "tnn": "Home_AP_SSID",
1360                    "cd": "Home_AP_PWD",
1361                    "wat": "WPA2_PSK",
1362                    "wet": "TKIP",
1363                    "swat": ["WPA_PSK", "WPA2_PSK"],
1364                    "swet": ["TKIP", "AES", "TKIP_AES"]
1365                  },
1366                "schema": { "$ref": "#/definitions/WiFiConf" }
1367              }
1368            }
1369          },
```

```
1370          "post": {
1371            "description": "Deliver Wi-Fi AP's information for an unboxing device to connect to it.\n",
1372            "parameters": [
1373              {"$ref": "#/parameters/interface-all"},
1374              {
1375                "name": "body",
1376                "in": "body",
1377                "required": true,
1378                "schema": { "$ref": "#/definitions/WiFiConfUpdate" },
1379                "x-example":
1380                  {
1381                    "tnn": "Home_AP_SSID",
1382                    "cd": "Home_AP_PWD",
1383                    "wat": "WPA2_PSK",
1384                    "wet": "AES"
1385                  }
1386              }
1387            ],
1388            "responses": {
1389              "200": {
1390                "description" : "",
1391                "x-example":
1392                  {
1393                    "rt": ["oic.r.wificonf"],
1394                    "if": ["oic.if.rw", "oic.if.baseline"],
1395                    "tnn": "Home_AP_SSID",
1396                    "swmt" : ["A", "B", "G"],
1397                    "swf": ["2.4G", "5G"],
1398                    "cd": "Home_AP_PWD",
1399                    "wat": "WPA2_PSK",
1400                    "wet": "AES",
1401                    "swat": ["WPA_PSK", "WPA2_PSK"],
1402                    "swet": ["TKIP", "AES", "TKIP_AES"]
1403                  },
1404                "schema": { "$ref": "#/definitions/WiFiConf" }
1405              }
1406            }
1407          }
1408        }
1409      },
1410      "parameters": {
1411        "interface-all" : {
1412          "in" : "query",
1413          "name" : "if",
1414          "type" : "string",
1415          "enum" : ["oic.if.rw", "oic.if.baseline"]
1416        }
1417      },
1418      "definitions": {
1419        "WiFiConf" : {
1420          "properties": {
1421            "rt" : {
1422              "description": "Resource Type of the Resource",
1423              "items": {
1424                "enum": ["oic.r.wificonf"],
1425                "type": "string",
1426                "maxLength": 64
1427              },
1428              "minItems": 1,
1429              "uniqueItems": true,
1430              "readOnly": true,
1431              "type": "array"
1432            },
1433            "tnn" : {
1434              "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
1435              "pattern": "^.*$",
1436              "type": "string"
1437            },
1438            "swmt" : {
1439              "description": "Indicates supported Wi-Fi mode types. It can be multiple",
```

```
1440              "items": {
1441                "description": "Supported Wi-Fi Mode Type.",
1442                "enum": [
1443                  "A",
1444                  "B",
1445                  "G",
1446                  "N",
1447                  "AC"
1448                ],
1449                "type": "string"
1450              },
1451              "readOnly": true,
1452              "type": "array"
1453            },
1454            "wat" : {
1455              "description": "Indicates Wi-Fi Auth Type",
1456              "enum": [
1457                "None",
1458                "WEP",
1459                "WPA_PSK",
1460                "WPA2_PSK"
1461              ],
1462              "type": "string"
1463            },
1464            "n" : {
1465              "$ref":
1466  "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
1467  schema.json#/definitions/n"
1468            },
1469            "swat" : {
1470              "description": "Indicates supported Wi-Fi Auth types. It can be multiple",
1471              "items": {
1472                "description": "Indicates Wi-Fi Auth Type",
1473                "enum": [
1474                  "None",
1475                  "WEP",
1476                  "WPA_PSK",
1477                  "WPA2_PSK"
1478                ],
1479                "type": "string"
1480              },
1481              "readOnly": true,
1482              "type": "array"
1483            },
1484            "swf" : {
1485              "description": "Indicates Supported Wi-Fi frequencies by the Enrollee. Can be multiple.
1486  Valid values are ('2.4G', '5G')",
1487              "items": {
1488                "pattern": "^(2\\.4|5)G$",
1489                "type": "string"
1490              },
1491              "readOnly": true,
1492              "type": "array"
1493            },
1494            "swet" : {
1495              "description": "Indicates supported Wi-Fi Encryption types. It can be multiple",
1496              "items": {
1497                "description": "Indicates Wi-Fi Encryption Type",
1498                "enum": [
1499                  "None",
1500                  "WEP_64",
1501                  "WEP_128",
1502                  "TKIP",
1503                  "AES",
1504                  "TKIP_AES"
1505                ],
1506                "type": "string"
1507              },
1508              "readOnly": true,
1509              "type": "array"
```

```
1510            },
1511            "wet" : {
1512              "description": "Indicates Wi-Fi Encryption Type",
1513              "enum": [
1514                "None",
1515                "WEP_64",
1516                "WEP_128",
1517                "TKIP",
1518                "AES",
1519                "TKIP_AES"
1520              ],
1521              "type": "string"
1522            },
1523            "cd" : {
1524              "description": "Indicates credential information of Wi-Fi AP",
1525              "pattern": "^.*$",
1526              "type": "string"
1527            },
1528            "id" : {
1529              "$ref":
1530    "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
1531    schema.json#/definitions/id"
1532            },
1533            "if" : {
1534              "description": "The OCF Interfaces supported by this Resource",
1535              "items": {
1536                "enum": [
1537                  "oic.if.rw",
1538                  "oic.if.baseline"
1539                ],
1540                "type": "string",
1541                "maxLength": 64
1542              },
1543              "minItems": 2,
1544              "uniqueItems": true,
1545              "readOnly": true,
1546              "type": "array"
1547            }
1548          },
1549          "type" : "object",
1550          "required":["swmt", "swf", "swat", "swet", "tnn", "wat", "wet"]
1551        },
1552        "WiFiConfUpdate" : {
1553          "properties": {
1554            "wat" : {
1555              "description": "Indicates Wi-Fi Auth Type",
1556              "enum": [
1557                "None",
1558                "WEP",
1559                "WPA_PSK",
1560                "WPA2_PSK"
1561              ]
1562            },
1563            "cd" : {
1564              "description": "Indicates credential information of Wi-Fi AP",
1565              "pattern": "^.*$",
1566              "type": "string"
1567            },
1568            "wet" : {
1569              "description": "Indicates Wi-Fi Encryption Type",
1570              "enum": [
1571                "None",
1572                "WEP_64",
1573                "WEP_128",
1574                "TKIP",
1575                "AES",
1576                "TKIP_AES"
1577              ]
1578            },
1579            "tnn" : {
```

```
1580              "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
1581              "pattern": "^.*$",
1582              "type": "string"
1583          }
1584        },
1585        "type" : "object",
1586        "required":["tnn", "wat", "wet"]
1587      }
1588    }
1589 }
1590
```

A.4.5   Property definition

Table A.6 defines the Properties that are part of the "oic.r.wificonf" Resource Type.

Table A.6 – The Property definitions of the Resource with type "rt" = "oic.r.wificonf".

Property name	Value type	Mandatory	Access mode	Description
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.
cd	string	No	Read Write	Indicates credential information of Wi-Fi AP.
wat	string	Yes	Read Write	Indicates Wi-Fi Auth Type.
swat	array: see schema	Yes	Read Only	Indicates supported Wi-Fi Auth types. It can be multiple.
tnn	string	Yes	Read Write	Indicates Target Network Name (SSID of Wi-Fi AP).
wet	string	Yes	Read Write	Indicates Wi-Fi Encryption Type.
id	multiple types: see schema	No	Read Write	
rt	array: see schema	No	Read Only	Resource Type of the Resource.
swmt	array: see schema	Yes	Read Only	Indicates supported Wi-Fi mode types. It can be multiple.
swf	array: see schema	Yes	Read Only	Indicates Supported Wi-Fi frequencies by the Enrollee. Can be multiple. Valid values are ("2.4G", "5G").
n	multiple types: see schema	No	Read Write	
swet	array: see schema	Yes	Read Only	Indicates supported Wi-Fi Encryption types. It can be multiple.
wat	multiple types: see schema	Yes	Read Write	Indicates Wi-Fi Auth Type.

cd	string	No	Read Write	Indicates credential information of Wi-Fi AP.
tnn	string	Yes	Read Write	Indicates Target Network Name (SSID of Wi-Fi AP).
wet	multiple types: see schema	Yes	Read Write	Indicates Wi-Fi Encryption Type.

A.4.6 CRUDN behaviour

Table A.7 defines the CRUDN operations that are supported on the "oic.r.wificonf" Resource Type.

Table A.7 – The CRUDN operations of the Resource with type "rt" = "oic.r.wificonf".

Create	Read	Update	Delete	Notify
	get	post		observe